An Argus Specialist Publication
JANUARY 1985
85p

NEW

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

Compunet calling – Commodore's new modem puts you in touch

Artistically inclined? We grapple with a variety of graphics packages

Games and utilities to wrestle with

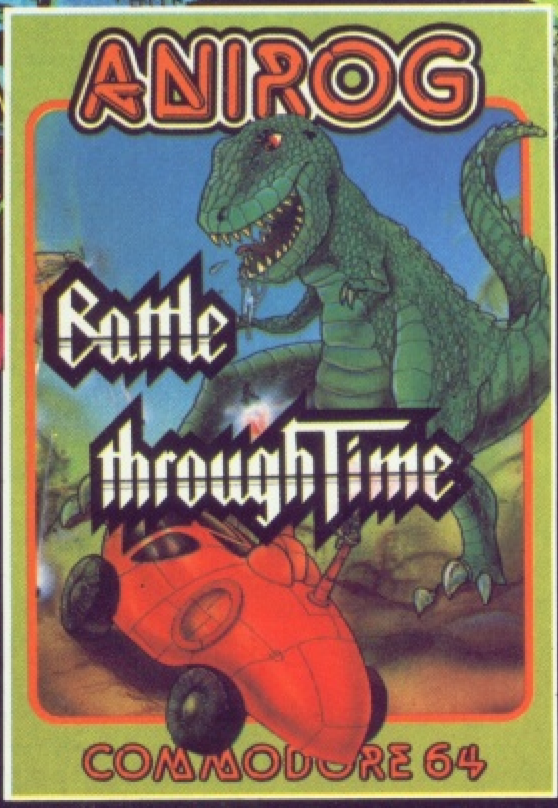
Add spice to your games with our great new series

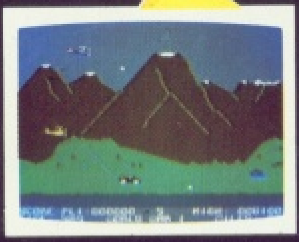
Software and peripherals – read before you buy



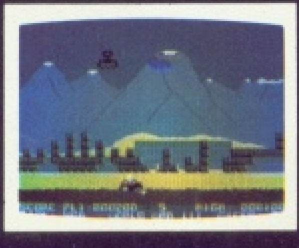
The new Plus 4 revealed











ANIROG

Trade Enquiries: **ANIROG SOFTWARE 10 VICTORIA IND. PARK, VICTORIA ROAD, DARTFORD, KENT. (0322) 92513/8**
Mail Order: **8 HIGH STREET, HORLEY, SURREY 24 HOUR CREDIT SALES HORLEY 02934 6083**
Payment by: **P.O. - ACCESS - VISA - 50p POSTAGE & PACKAGING**

In the year 2525 A.D., time travel is finally perfected. Scientists of the project send you into the time void equipped with an armed, all terrain vehicle. Your objective is to assist mankind in his various battles : from the prehistoric age to the aftermath of the final conflict. Only a successful mission can assure your return. Features include 'Hi-Score table, two player option, two skill levels and continuous play facility.

Commodore 64 £7.95

Our COMMENT

Welcome to this month's issue of Your Commodore. The editor considers Compunet and offers the compliments of the season.

HOME COMPUTING HAS long been regarded by the uninitiated as a solitary pursuit. We are supposed to sit for days on end, without food or drink, in semi-darkened rooms huddled over the keyboard with only the strange glow of the screen to keep us company. It can happen, of course, though I have yet to go for a period of longer than two hours without some form (usually liquid) sustenance. All the signs point in the opposite direction but have you tried convincing people? Show them the two joystick ports on the 64 and point to the increasing number of games which require the use of both. Tell them about the growing number of computer clubs and their ever increasing membership. Show them that programs are written for useful purposes and not for their own sakes; indeed tell them that many programs, and not just commercial ones, are written by teams of people, someone designing the graphics, someone else the music and so on. All to no avail.

The answer could have arrived in the shape of Compunet. At last Commodore owners can communicate with each other through the medium of their computers. By attaching the modem to your CBM 64 you could be at the forefront of a whole new form of communication. Of course, viewdata (of which Compunet is another manifestation) is not new. Prestel and its offshoot Micronet 800 have been with us for some years and it is also true that neither have really fulfilled their initial promise. Prestel suffers from a debilitating identity crisis: is it aimed at the business or the domestic user? It began as a method of using all that spare telephone capacity during off-peak times but it was soon adopted by diverse sections of the business community. In order to repair the balance Micronet 800 was started. Directed at the home micro user it offered an innovative and well presented package (and still does) but it is very one sided — Micronet provides the information in the form of news, information and software and the user downloads it.

Compunet offers the revolutionary advance of a two-way service. The user can up-load his own software as well as down-loading other people's. You can find out more in the article on the new service in this issue. Let me just finish by saying that Compunet offers Commodore owners the chance to dispel forever the myth of the computer hermit.





The Plus/4

The other big recent launch from Commodore is, of course, the new machines. Last month we looked at the VIC's replacement, the C16 and word is already coming through that it's selling extremely well. At the price and with its specifications this is not at all surprising. The Plus/4 which we look at this month is rather different. The market does not already exist for it. Instead it will have to create its own.

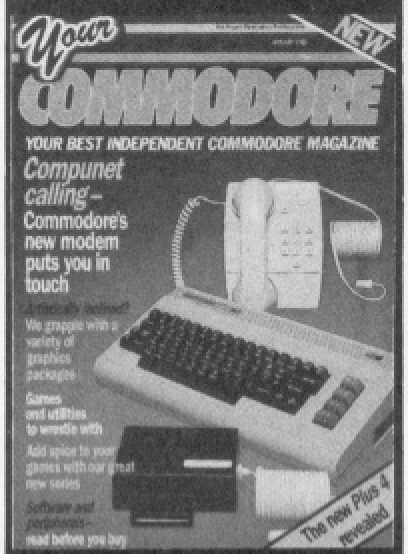
The idea behind the Plus/4 certainly looks promising and we shall take a look at the built-in software next month.

For the moment the hardware is considered by our reviewer who is not unimpressed.

All the best

This issue also contains a large number of graphics related articles. The Commodore machines are all renowned for the excellent graphic capabilities but there is also room for improvement. You can choose the solution which most suits you by reading our reviews of lightpens, of Logo and the Turtle, of the Koala Pad and of many of the Commodore drawing packages.

So whatever your interest there is something in this issue of Your Commodore for you. And strange as it may seem (and it often confuses us although this is the January issue you should be reading this before Christmas: so from all of us at Your Commodore may we wish you a very merry Christmas and a Commodore filled New Year.



NUMBER 4
JANUARY 1985
Editor: Wendy J Palmer
Deputy Editor: Kevin Cox
Editorial Assistant: Alison Hjul
Advertisement Manager: Mike Segrue
Advertisement Copy Control: Laura Champion
Chairman: Jim Connell
Origination: Ebony Typesetiing
Design: MM Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB
Telephone: 01-437 0626
Telex: 8811896

Your Commodore is a monthly magazine appearing on the first Friday of each month.

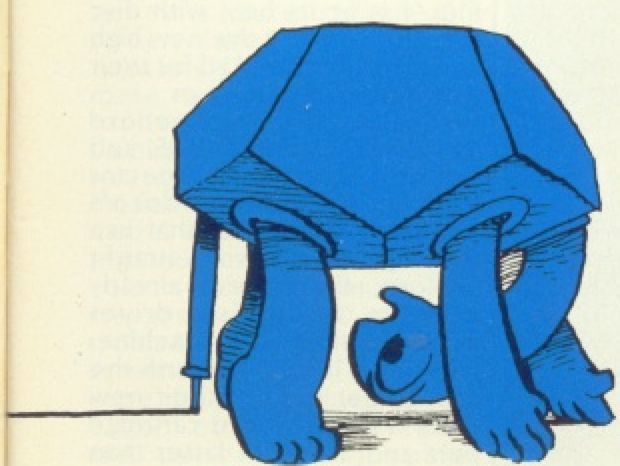
Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS.
Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.





CONTENTS





The Magic Roundabout
CRL
FOR THE ZX SPECTRUM 48K



BEYOND
CHALLENGING SOFTWARE
ANKH
Commodore 64

Following the review of the Commodore 16 last month, Mike Roberts places its big brother, the Plus/4, under the magnifying glass.

PLUS/4: HOW DOES IT ADD UP?



THE COMMODORE PLUS/4 IS Commodore's first entry into the top level home/very small business bracket. Commodore said that it wasn't suitable for running a business but went on to say that it could be used in certain applications by a 'professional' person. This is one of the most honest statements that I have heard a computer company say on the subject of the business use of a home computer. Clive Sinclair once said that you could run a power station on a ZX80 (1K RAM, 4K ROM no keyboard!)

Where this business subject comes from is the four built in software packages — wordprocessor, spreadsheet, database, and graphics — hence Plus/4, but more about this is a later article.

Externally the Plus/4 is a gunmetal, wedge shaped box with a keyboard taking up most of the room on the top and ports and interfaces taking up the whole of the back and sides. Now it becomes obvious why Commodore opted for the smaller type of connector instead of keeping it the same as they have for ages. If the normal 'chunky' type edge connectors and D sockets were used there would not be enough room around the edge of the machine.

The magic touch

The keyboard excels even Commodore's beautiful quality although some people express reservations when they first get their dabble with it. However, after a period of use everybody loves it.

The keyboard is very similar to the SX-64, and is angled, sculptured and curved. This makes typing on it a real dream. The layout consists of 68 keys, 60 of which are normal and 4 function key above the top row and four cursor control keys arranged in a cross shape. One odd thing about these keys is the shape, they look like arrows — the way the arrow is pointing indicates the direction that the cursor will go in.

The ports around the back and sides of the box show a departure from the 64/VIC stable. The VIC and 64 had almost identical I/O.

All change

Most things, however, have been changed; the cartridge/expansion port has been reduced in size to stop people shoving CBM 64 cartridges into it. I do not know whether the highly advanced structure of the CBM 64's slot with its facility for second processors is duplicated on the Plus/4. But given the nature of the machine as a small systems/serious device, this is more than a distinct possibility.

The Plus/4 support sideways ROMs; these were first implemented on the BBC micro some years ago and computers are starting to get these featured. Commodore call them 'Function key packages'. The reason for this rather strange name is that when the machine boots up, it checks to see if there are any ROMs attached. If there are then it will assign each ROM to a function key. The internal software uses this system so pressing F1 gets you into it. If the ROMs are removed the function key goes back to its original state. External ROMs will go on F2. It is unknown how many ROM cartridges can be attached to the machine at once, but there is the possibility of four (BASIC, the in-built ROMs, plus two others).

The two D9 connectors of the CBM 64 have been dispensed with and replaced with mini DIN connectors. This means you can only buy Commodore's joysticks. Commodore joysticks are not the best things in the world, even their new 'hi-tec' style ones.

The Cassette recorder socket is also a mini DIN connector, this is because the C16 cassette deck is different to the old tape decks. The Plus/4 is at its best with disc drives, including the new high speed ones intended for their new range of machines.

Thankfully, Commodore have left the Serial BUS, and the audio/video connector alone. All Commodore's existing peripherals, that use these ports, will work straight off, so there are already printers and disc drives available for the machine. However, it may be worth the wait to get hold of their new discs which use the cartridge port and are a lot faster than the old ones.

Inside knowledge

Moving on to the internal hardware reveals some surprises. Most of the insides is driven via one big chip. Called either the 7501 or the TED chip depending on your inclination, it combines a 6510 processor at 2MHz with a sound generator, timers, input/output, memory banking, and graphics generation. In all it has 19 registers to control things.

Sound quality is as good as the next man's although it only has two channels. These can be two sound channels or one sound and one noise (for special effects). Nearly all the advanced sound features of the SID chip have been left out like ADSR, filtering, and modulation.

Graphics ability is superb. It is natural that this will be compared with the Commodore 64 as there are a lot of similarities in spec. However, the graphics are different and there are currently two schools of thought as to which is better the CBM 64 or the Plus/4.

Simulated sprites

The big difference is sprites. These wonderful things that make games programming easy have been chopped from the Plus/4. In their place is a software simulation of them from BASIC where you can extract an area of the screen and store it in a string. This string can then be recalled and put back on the screen at any point. There are also other options to manipulate these objects, but they are not true sprites; a large 120 byte object takes about a quarter of a second to write to the screen. I feel that the world can live without sprites, for at least another computer generation (about 18 months); the Commodore 64 and Atari were just too far ahead of their time.

The trade-off against the sprites is more colour. The screen of the C16 can have 128 colours (121 excluding black) made up of 16 colours, 8 luminence levels, and flashing. Screen size is 40 x 25 text with four other graphics modes. The other graphics modes are 320 x 200 with the previously mentioned 128 colours being used in a colour map system, and 160 x 200 in a multicolour form. Both hires screens have an option to leave four text lines at the bottom of the screen. There are some other graphics modes and options but these are only available by POKEing. User defined graphics (UDGs) are obtained by POKEing and manipulation of registers.

The manual gives no hint of these although they are very straightforward to obtain. When playing with UDGs one other feature becomes apparent. A character generator is 2K long, (256 x 8 bytes) the C16 one is only 1K long, how come? Well, the long and short of it is that the C16 uses a hardware reverse field attribute. The top bit of the current character displayed indicates whether it is inverted or not. This has some advantages and disadvantages. The advantage is in memory consumption. The disadvantages are that you can only have 128 UDGs, and flashing works in a rather strange way. A reverse field space is shown as a black square. When you flash it instead of getting a flashing square nothing happens. This is quite confusing until you realise that a flashing space doesn't change.

Other features

Other modes not documented include Extended Background Colour mode, which gives you different background colours as well as foreground colours, and multicolour characters where each character can be made up out of a number of colours. There may be others but I must wait until I get a technical manual to find them out.

This brings me onto another point. The BASIC is ideal for an inexperienced user or an experienced BASIC user, but what about us machine code hacks and people who wouldn't use BASIC if they were paid to?

The answer is TEDMON. This is a full feature assembler, disassembler, monitor, debugger. It is similar to Extramon 7.5 and is very good indeed. This makes writing assembly language very easy as you already have most of the development software built in.

The monitor can also be called by using the reset button. This is a great feature and is in a little recess just by the power supply. Press it in and the machine goes back to its power on state — memory contents are preserved but it is awkward to get at them. The beauty of it all is when you keep the STOP key pressed down at the same time as you press in the reset key. The computer jumps into the monitor, key in 'X' (for eXit) and you are back in BASIC, complete with intact program.

The manual is excellent and way past Commodore's usual standard. It is informative and instructional for the first time user. For the experienced person there are memory maps and register details.

In conclusion

The Plus/4 scores over its little brother the C16 by having 64K of memory, and now comes the good bit — the BASIC has built in memory banking so that you can use the extra memory to the full. When using hi-res this only cuts you down to 50K. Compare that to a BBC that leaves you with a fifth of that after hi-res has taken its chunk out. This mammoth (sorry elefantine!) memory will mean that there should be a lot of huge adventures and great arcade games on the market to use these.

The BASIC is identical to the C16 and all I said last month applies here. This is the computer of now, excellent BASIC, keyboard, software built in, and best of all a vast amount of user memory. The problem is whether it is worth it at the price — £299. This is well below the BBC with which it is comparable. But then the BBC always was over priced. Only time will tell, as the Plus/4 is carving a new area in the market.



ATTENTION! ATTENTION! ALL COMMODORE USERS!

A FREE LIGHTPEN!

Yes a FREE Light Pen! with every **GRAPHKIT!** Graphkit is the ultimate in drawing!, designing! or painting! (using the light pen of course, which incidentally has a full 3 year warranty!) Graphkit will amongst other things allow you to:

- Draw free handed on the screen anything from a Micky Mouse to your latest integrated circuit! With all the fill colour and erase facilities of course!
- Save/load your latest master piece to/from the disk/tape drive
- Or directly copy the drawing on the screen to the printer! And keep it for ever or send to a friend as a post card!
- Design your own colourful sprites and characters
- Play the games in the package! or use the light pen in your own games/education/programs

GRAPHKIT is available on disk and tape. Tape version is £1595 and disk version (recommended) is £19.95.



All you 1540/1541 disk users **DMON!** Is here at last. Dmon is the disk monitor you have been waiting for it will allow you to:

- read/write blocks from/to the disk
- display and edit blocks on the screen
- display and send disk messages/commands
- transfer your programs from tape to disk or disk to disk or even disk to tape! ● and more!

DMON comes on DISK at only £9.95.

TORNADO 20/64

Do you use tapes? Are you sick and tired of waiting for your programs to be Loaded/Saved from/on the tape? Then you need a **TORNADO!!** Tornado allows you to Save/Load/Verify your Basic/machine code programs faster than a CBM 1541 disk drive does! Due to popular demand! Tornado now comes with new and more powerful commands plus extra instructions to assist you in making fast versions of your existing machine code/Basic programs. Tornado is available on tape for CBM 64 and 8K + Vic 20.

A FREE BREAKER WITH EVERY TORNADO IF YOU USE THE COUPON BELOW NOW!

BREAKER 20/64

Do your Run/Stop and Restore keys often fail? Do you want to come out of those crashes?! Or get into those unbreakable programs?! Then what are you waiting for, get yourself a **BREAKER!!** Reset switch and let your computer know who is the boss!! Breaker can be connected to your machine in seconds, no soldering. Included with the Breaker is a copy of basic recovery software on tape. Now available for any CBM 64 or Vic 20.

YC/12/84

BSF CSL, 82 FURNACE DRIVE, CRAWLEY, W SUSSEX RH10 6JE.

Please send me ....Tornado at £9.95 + my free! Breaker

Please send me ....Breaker at £7.95

Please send me ....Light Pen at £11.95

Please send me ....Graphkit at £15.95 (tape)+ my free LP

Please send me ....Graphkit at £19.95 (disk)+ my free LP

Please send me ....DMON at £9.95

I enclose cheque/PO/cash to the sum of £..................

Name..........................Computer..........................

Address..

....

Please use this form and block letters only. All prices are inclusive of p&p (mail order only, add £2 if outside UK). Please post to:
BSF CSL, 82 FURNACE DRIVE, CRAWLEY, WEST SUSSEX RH10 6JE.

DATA STATEMENTS

Commodore in the classroom

COMMODORE WILL BE sponsoring the prestigious 1985 British Computer Society Schools' Computer Quiz. It commited itself to increasing the number of computers in schools in 1984 and sees this quiz as a vehicle for its education offensive next year. Commodore has contributed sponsorship of over £50,000 including prizes, for schools worth over £12,500.

In conjunction with the quiz, Commodore is running a special schools' loan and sponsorship scheme. Every school taking part in the quiz will be offered the free loan of two complete Commodore 64 computer systems for three months and, if at the end of the loan period the school wishes to keep the loaned systems, Commodore will match every £7.00 the school raises with £3.00.

The quiz comprises a quiz master and six to eight team and individual rounds of questions. Schools in each area will be asked to select a team of three pupils — one under 17 years, one under 16 years and one under 15 years. Local area heats start in November and the 40 schools thus emerging as winners of the branch finals will each receive a Commodore communications modem for use with Compunet, Commodore's new on-line database service. The 40 branch finalists will then go on to compete in eight regional finals to win prizes of Commodore 64 computers and 1541 disc drives, in total worth over £5,500.

The eight regional finalists will then compete in a national final scheduled for July 1985. The winning school will receive £1,600 worth of Commodore computers and peripherals, and the runner-up will receive Commodore equipment worth £700; there will also be many different prizes for individual team members.

Commodore plan to folow each regional final with a 'roadshow' giving parents, teachers, children and local people the chance to try out Commodore's range of computers along with their educational and recreational software and the new modem.

Further details are available from Commodore Education Department on 0536-205252.



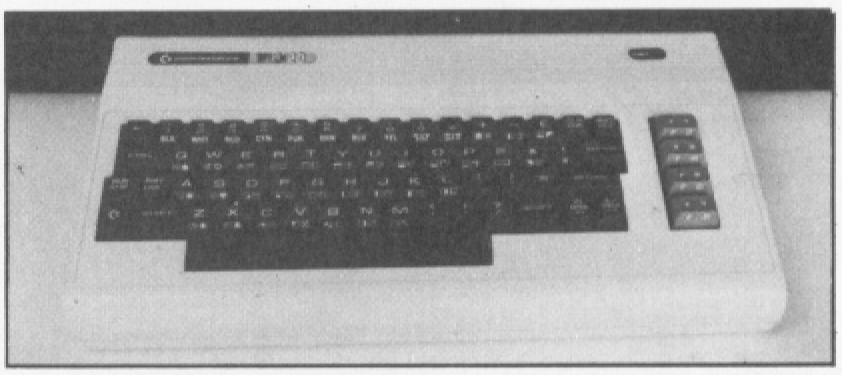
VIC 20 word processing

ATLANTIC SOFTWARE HAVE released their TOTL range of word processing programs for the VIC 20.

TOTL. TEXT.2.0, for the VIC 20 with at least 8K expansion claims to offer full formatting control, text input, editing and insertion in a single mode, with full cursor control, forwards and backwards scrolling and direct screen editing. This package can be used not only with Commodore printers but also with suitably linked parallel and RS-232 interface printers. TOTL.TEXT.2.0. is menu-driven and incorporates machine code sections to speed loading and printing and may be used to print multiple copies from memory, or documents of any length from files.

TOTL.TEXT.2.5 is provided for the VIC 20 with 16K and contains extra features such as embedded footnotes, enhanced characters and printing direct from tape or disc files. Direct keyboard input while printing makes this program suitable for mailshots and form letters.

The software is available on tape and disc and prices range from £11.95 to £19.95. TOTL software is available from Atlantic Software, 28 Park Farm Road, Kingston, Surrey, KT2 5TQ.



Norbain interface

NORBAIN MICRO LIMITED have launched their Turbopoint/GT, a printer interface for the Commodore 64 and VIC 20, following an exclusive UK distribution agreement with the American manufacturer, Telesys Computer Peripheral Products.

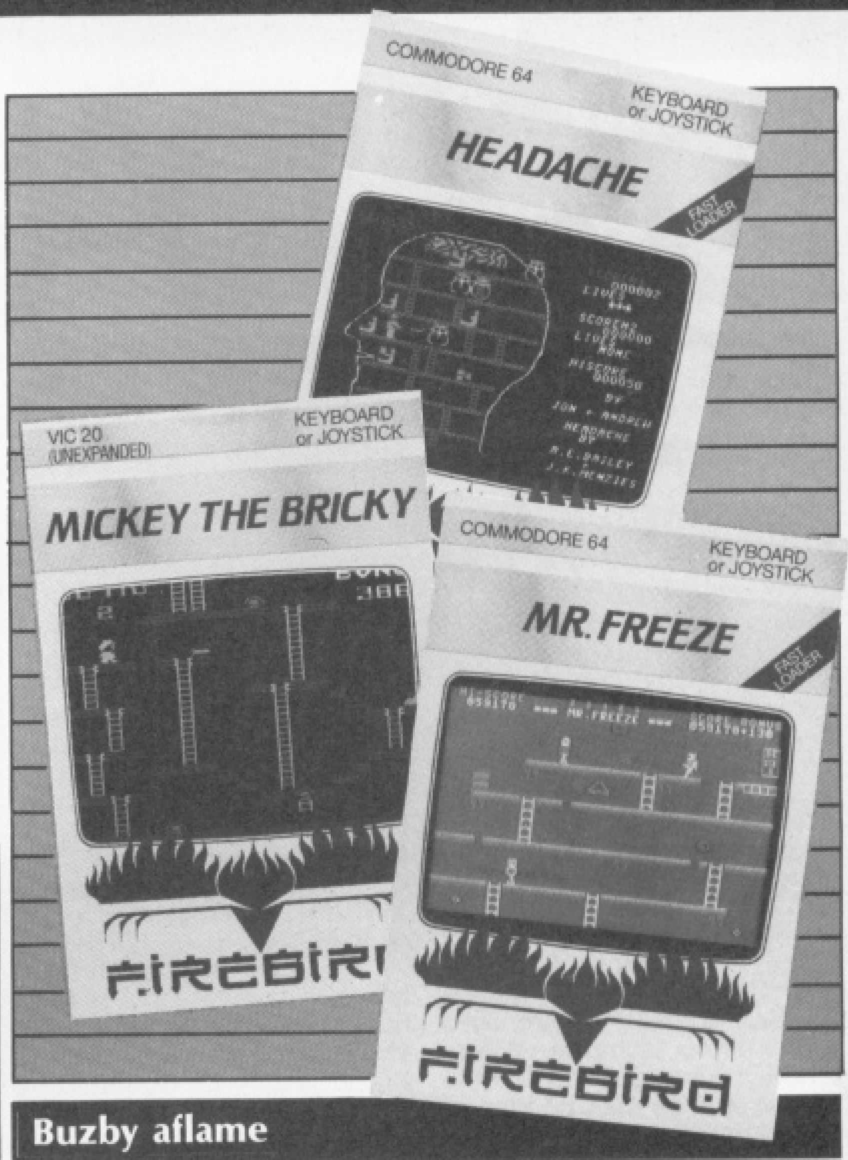
This interface enables the Commodore 64 and VIC 20 to be connected to most of the popular makes of printer currently available on the market. It features a DIP switch selector for different printers as well as the printing of enhanced Commodore graphics including reverse characters, a very fast graphics dump and a special line buffer which doubles the text printing speed on printers without onboard memory.

The Turbopoint/GT interface also includes an optional plug-in 16K printer buffer to overcome the low print speed of the printer when transferring data to the printer at the computer's maximum speed.

The Turbopoint/GT printer interface sells for £65.00. Norbain Micro Ltd. can be contacted at Norbain House, Boulton Road, Reading, Berkshire, RG2 0LT; telephone 0734-752201.



DATA STATEMENTS



Dream makers

WILDEST DREAMS SOFTWARE Rental are working on the premise that if you can rent video films why not computer games? This company, from November 9th, will be offering a package of new Commodore games tapes for rental only. These tapes will be available for hire only through video dealers.

Wildest Dreams can be contacted at P.O. Box 84, Coventry telephone 0203-663085.

The Ultimate Game

IF YOU'VE EVER SEEN ONE OF Ultimate Play The Game's programs you'll know why Spectrum owners rave about them. Atic Atac and Sabre Wulf are both now classics. Unfortunately, CBM 64 owners have been deprived — until now. The Staff of Karnath is their first game for the 64 and it should be out in time for Christmas. It is an arcade adventure in the style of their most recent games and going on past form the graphics should be breathtaking.

Ultimate Play the Game, The Green, Ashby de la Zouch, Leicestershire, LE6 5JU.

One for the road

CRL HAS ALSO RECEIVED approval from the AA for its Commodore version of 'Highway Code'. This program consists of over a hundred questions, all of which can be found within the book produced by HMSO.

The user chooses either a 25 question test or ten question test from which he/she is given a percentage rating. The program includes a number of graphics which represent road signs and traffic situations. 'Highway Code' will cost £5.95.

CRL can be contacted at CRL House, 9 Kings Yard, Carpenter's Road, London E15

Buzby aflame

BRITISH TELECOM'S NEW software house is producing a new range of low-price home computer games under the label Firebird. Mr. Richard Hooper, Chief Executive of British Telecom's Value Added Systems and services, hopes that these games, which retail at £2.50, "...will offer the same quality as some products costing twice as much." Five Commodore and two VIC 20 titles are included in the initial release. These are all games, but Mr. Hooper hopes to eventually offer "... educational software and other types of programs".

Eastern promise

DOMARK HOPE THAT A £25,000 prize will do for 'Eureka' what million pound bingo prizes have done for Fleet Street sales, since they expect it to be a number one hit by Christmas.

Eureka comprises five complete adventure stories in one. With a plot unfolding across five time zones from pre-history to the present day. The five adventures can be enjoyed independently and each contains vital clues that lead to the £25,000 reward. Each adventure starts with a mission and your success in the adventures is aided by your skill as a warrior.

Eureka was programmed by Hungarian software company, Andromeda. It took the equivalent of five years to create along with the skills of four graphic artists, two musicians and a professor of logic.

The program is accompanied by a booklet of riddles and verse which, together with the game, leads to a mystery telephone number somewhere in Britain.

Eureka, which has already been translated into French with a Spanish translation underway, is available on the 64 at £14.95 for the tape version and £16.95 for the disc version.

Domark can be contacted at 228 Munster Road, London SW6.



Ghostbusters

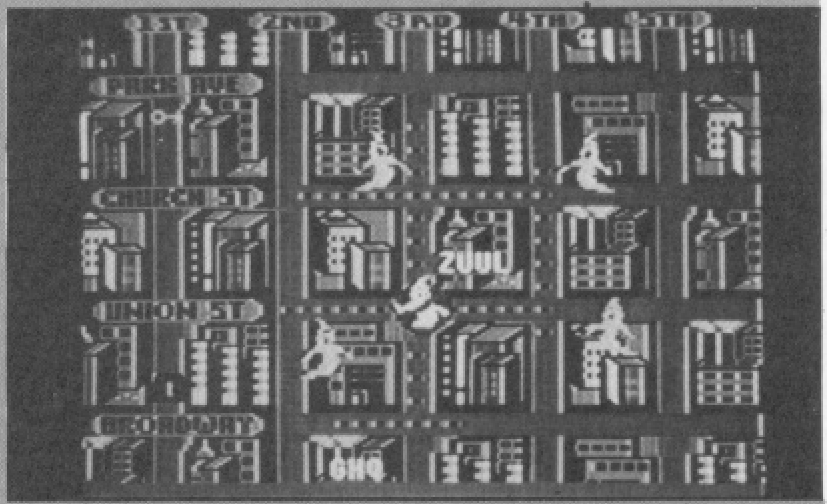
WE WERE SPIRITED OFF TO the Premier Theatre in Shaftesbury Avenue on Halloween night for a sneak preview of Colombia Pictures 'Ghost Busters'. Already bored silly by Ray Parker Jr's song of the same name, I faced the evening with trepidation. But I was in for a surprise. With it's 'good clean fun' approach, I can understand how this outstanding piece of family entertainment is achieving such success in America. The film, which tells the story of three New York parapsycholgists who set up a ghost-fighting business, has become the all-time box office record for Columbia Pictures.

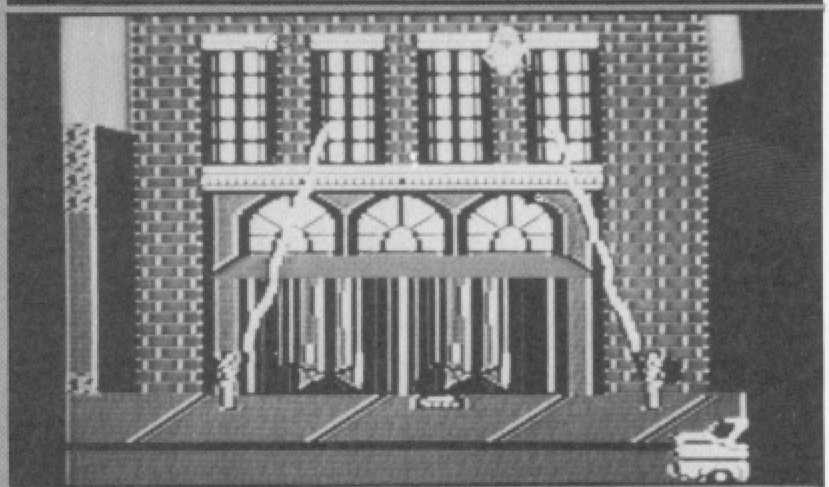
Our evening's free entertainment was at the invitation of Activision who have acquired from Columbia Pictures the exclusive worldwide rights to develop Ghostbusters home computer software. The game claims to follow the film closely, using the music and characters from Ghostbusters.

As with the film, the game requires you to fend off a ghost invasion of New York and make sure your Ghostbusters make it to a final dramatic showdown at the Temple of Zuul. With the cash you are given to build a Ghostbuster franchise, you choose and equip your own vehicle and watch your debits and credits rising or falling, depending on your skill as a Ghostbuster. With all sorts of gadgets at your fingertips, such as an Energy Detector, Marshmallow Sensor or ghost Vacuum, you must search for, catch and store ghosts. New York street maps are provided to guide you on your mission.

Your progress is checked by a Status Report and, if you're successful enough and earn enough money, the Ghostbusters franchise and secret number you acquire can be used next time you play. Your own secret number may be seen on any version of Ghostbusters anywhere in the world.

The game is now available on the Commodore 64 at £10.99 for the tape version and £19.99 for the disc version.







At Argus nobody can hear you scream...

COMPUTER GAMES BASED on box-office hits seem to be all the rage. Argus Press Software will be releasing their game, Alien, on 19th November to pre-empt the relauch of the sci-fi horror movie on January 1st. The game will be on sale in over 250 cinemas across the South of England and W.H. Smith will hold a competition in the cinema on the opening night in each town; the prize will be a copy of the game.

Alien is the first of Argus' Mind Games to include PCS (Personality Control System) which sets up the personality variables for each game. The behaviour of each character in a particular situation is determined by specific instructions, the history of the personality and his/her experience in the game.

The point of the game is to try and beat the Alien while you are increasingly trapped in your spaceship, the Nostromo. You are aided in your struggle by trackers and other weapons, but have your characters got the courage to use them?

Alien is available on the Commodore 64 at £8.99.

DATA STATEMENTS

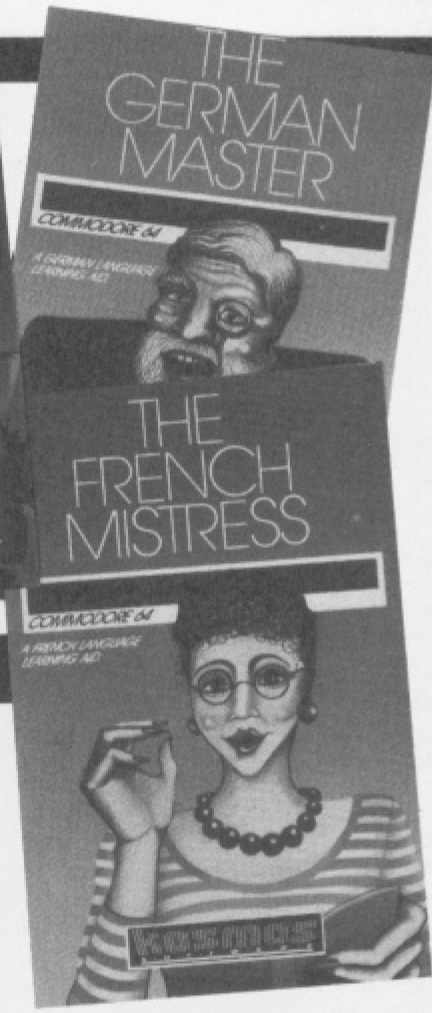
Stock market



KUMA COMPUTERS ARE hoping that, rather than splash out vast sums of money on computer games, that some Commodore users might need some assistance in investing their money. With their latest game, 'Stock Market', they have simulated the London Stock Exchange. Players are given news items on company shares and advice, hints and tips for likely good investments — although not all of these are accurate. Players win or lose money through placing and managing investments.

Stock Market can be played by 1 to 4 players and costs £6.95. Kuma Computers Ltd. can be contacted at 12 Horseshoe Road, Pangbourne, Berks, RG8 7JW; telephone 07357-4335.





Continental drift

KOSMOS SOFTWARE HAVE released Commodore 64 versions of their educational titles, The French Mistress, The German Master and The Spanish Tutor. The programs provide learning aids for thousands of foreign words, verbs and phrases.

Each package consists of a tuition control program and pre-recorded lessons, which can be used in a variety of learning modes and, finally, a translation test mode. Each language is covered by two cassettes (Level A and Level B) covering different areas of grammar and vocabulary.

Each cassette costs £8.95 (i.e. £17.90 per language) although Kosmos insist that the cassettes may be bought separately as each contains the necessary control program. Kosmos Software Ltd. can be contacted at 1 Pilgrims Close, Harlington, Dunstable, Beds LU5 6LX; telephone 05255-3942.

More Melbourne games

FOLLOWING THE SUCCESS of their 'The Commodore 64 Games Book', Melbourne House have released a sequel, 'Commodore 64 Games Book 2'. Melbourne House claim that the book contains 30 'completely new and original games listings' with a CHEXSUM listing supplied at the end of each program so the reader may determine immediately whether a program typed in contains a transcription error, in which case the line number where the mistake has occurred is pinpointed.

In keeping with the times, Melbourne House have also released the 'Commodore 16 Games Book', which includes a variety of games from educational and simulation games to gambling games. Once again, the CHECKSUM utility is supplied.

'Commodore 64 Games Book 2' retails at £6.95 and 'Commodore 16 Games Book' at £5.95. Melbourne House (Publishers) Ltd. can be contacted at Castle Yard House, Castle Yard, Richmond TW10 6TF; telephone 01-940-6064.



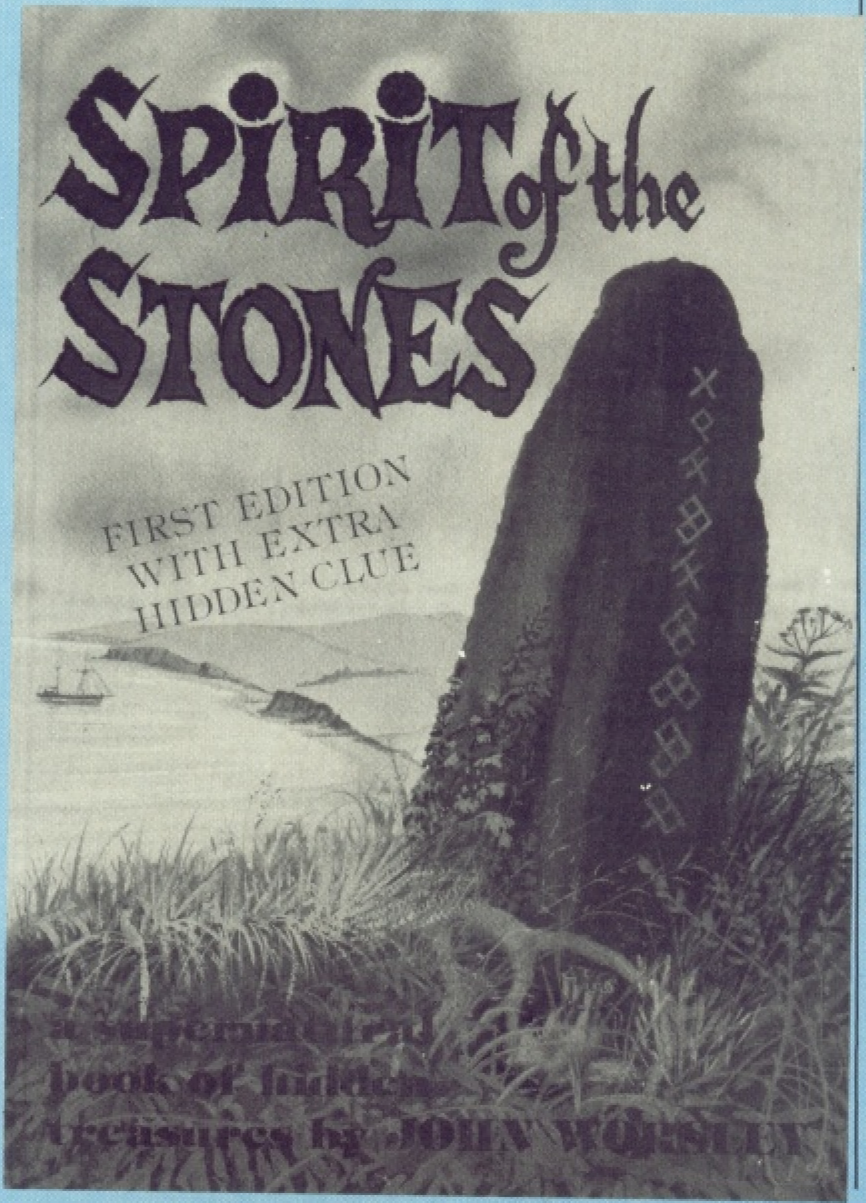
Scrolling Stones

SEA, SUN, SAND AND NOW software. The Isle of Wight has got the lot. Software? Commodore has just launched Spirit of the Stones, an arcade game with 21 locations based on a scrolling map of the island. However, as well as a game you also get the chance to join in a treasure hunt for 40 small talisman, each containing a single diamond and for the Great Wight Eye all of which have been scattered around the island. Fortunately, (or perhaps unfortunately) you don't have to take the ferry over to the island to claim the treasure.

Included with the game is the book of the Spirit of the Stones, a tale of smuggling and the supernatural which tells the tale of how the talismen came to be scattered. The book, written by island resident John Worsley, is also full of visual and verbal clues to the whereabouts of the stones and one of the first steps for all would-be treasure hunters must be the deciphering of the runes or secret writings which border the pages.

Those clever enough to unlock the key to this puzzle and find one of the talismen will also be eligible for a share of the Royalty Fund into which 50p will be deposited for every package sold. However, the share-out will not be until 1993 or when the £1 million ceiling is reached.

Secret of the Stones costs £14.99 and is available from Commodore.



Tymac Talkies

TYMAC HAVE RELEASED A series of 'talkies' for the Commodore 64 and VIC 20. These arcade games feature speech without the use of a synthesiser.

The games include Gandalf, where you fire power bolts at ruthless attackers, Pegasus and the Trials of Perseus where you must fight strong creatures while searching for treasures of the ancients in combat with the Samurai and code name: DEADZONE where, as the heroic lieutenant, you must battle with the mad leader of a hostile nation which has developed a deadly virus which he plans to unleash on the world. Also available from Tymac, for the VIC 20, is an educational/fun program called Type Snyper. Wizard's Graphics, a VIC 20 utility designed to generate high resolution mutli-coloured graphics quickly and easily on screen, should be available soon.

Tymac Talkies are designed by the American company Game Gems Inc. Tymac (UK) Ltd. can be contacted at Nettleton House, Calthorpe Road, Edgbaston, Birmingham, B15 1RL.

Four from Beyond

BEYOND HAVE ANNOUNCED the launch of four Commodore 64 titles — Ankh, Aztec, Mr. Robot and My Chess II.

Ankh is described as a "computer mind game...in which players explore a mysterious, maze-like, 'Metareal' world". Explore with your 'mindprobe' (a device to manipulate or destroy objects found on-screen). Within each section their lies a task, which must be solved through the mindprobe and creative thinking, a tool or treasure to collect for help in solving future puzzles, or a trap. Beyond reckon you will need assistance with this game and are, thus, providing a special helpline. They are also offering a prize to the first player to solve a particularly difficult problem within the game.

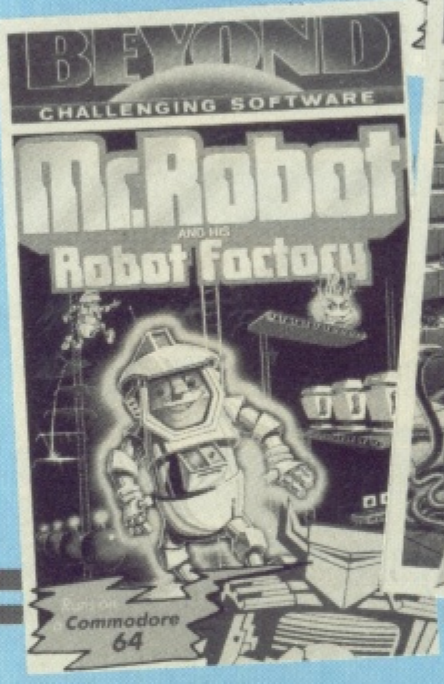
In Aztec, players control an 'Indiana Jones-style hero searching a forbidden temple for a priceless golden idol'. Each Aztec adventure is different and, as the game is loaded, the computer creates an eight level adventure setting from a library of 32 different floors.

In Mr. Robot, players use the Robot Factory to create and play their own screen designs on the A-side and guide Mr. Robot around a 22 screen obstacle course, collecting power pills and killing alien fires, on the B-side.

Beyond describe their Chess II, with nine levels of play and a default of 2, as 'the Rolls Royce of computer chess programs'. You battle against the computer in 3D with a view of the chess pieces from the rear and the side of the board.

All the games will be available on cassette at £8.95 or on disc at £11.95. Beyond can be contacted at Lector Court, 153 Farringdon Road, London EC1R 3AD.







DATA STATEMENTS

What? said Dylan



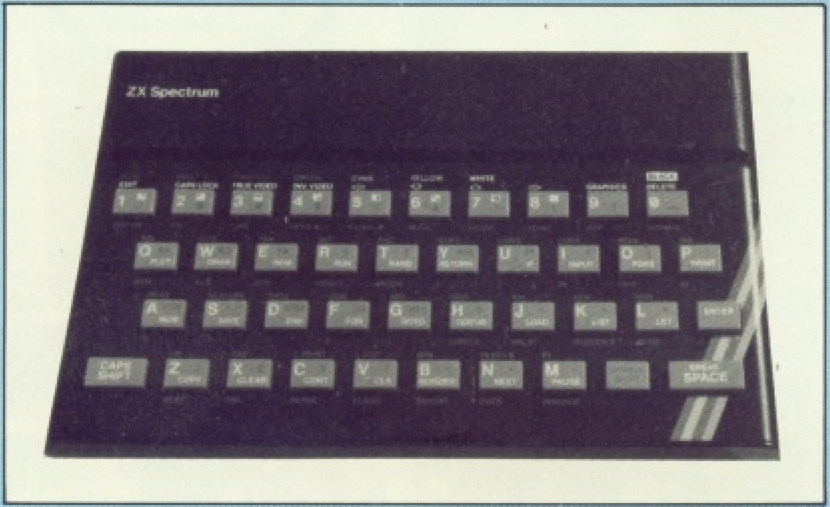
CRL, WHO HAVE ALREADY released the program 'The Magic Roundabout' for the 48K Spectrum, will be releasing the Commodore 64 version before the end of the year.

The program is based on the BBC's popular children's programme. The game takes place in and around the confines of the Magic Roundabout where Dougal is trying to build a sugar house. His fellow characters insist on eating the sugar cubes he needs to complete the construction as the sugar gives them all much needed energy.

Spectrum simulator

WHITBY COMPUTERS LTD. claim to have produced a Spectrum simulator for the Commodore 64. It sells for £14.95 and should allow you to use all Spectrum BASIC programs and some machine code programs on the 64 without modification. It should be available just after Christmas on turbo load cassette. We'll believe it when we see it!

Whitby Computers Ltd. can be contacted at 7 Chubb Hill Road, Whitby, North Yorkshire; telephone 0947-604966.



Commodore face the music

COMMODORE USERS CAN now turn their 64s into a musical instrument with a new music package from Commodore. The package is called 'Music Maker' and comprises software, a book containing 28 popular songs and tutorial material and a 24 key piano style keyboard which fits over the computer keyboard.

'Music Maker' has been specially developed for the Commodore 64 by Music Sales Ltd, the world's leading music publishers.

The software's eight built-in voices simulate instruments such as the guitar, piano and synthesiser; all the parameters of each sound can also be altered to create almost any sound within the capability of the Commodore 64. A tune can be created around a range of rhythms from waltzes to disco, with pre-programmed bass patterns. You can also use 'Music Maker' to create three note polyphonic music.

'Music Maker' caters for those Commodore users with no musical experience with its single step input mode which allows songs from the book provided to be entered into memory by note or letter. The rhythm is then 'tapped out' on any key and the completed tune played by the computer. Finally, the piano keys are duplicated on the screen and light up when the corresponding note is played.

Music Maker will sell for £29.95 and will be available on disc or cassette before Christmas.

Game of the Century?

JEREMY SAN, FOUAD DATAN and Andrew Glaister are responsible for Century Communications Ltd's new arcade game, Skyline Attack.

After taking off from London, the aim of the game is to pilot your plane across the world's cities, shooting aliens and scoring points while watching your fuel gauge. The game, for either one or two players, includes 18 different types of aliens, 10 levels and joystick or keyboard option.

A novel addition is the game Sssnake! which provides an alternative to staring at a blank screen while the program loads.

Skyline Attack is available on the Commodore 64 for £7.95. Century Communications is a division of Century Publishing Company Ltd. and can be contacted at Portland House, 12-13 Greek Street, London W1V 5LE; telephone: 01-434-4241.

Sterling down under

STERLING PUBLICATIONS Limited have entered the games market through a newly-formed division, Sterling Software, whose first titles include the winner of the Sterling Software competition Orpheus in the Underworld.

Orpheus is an arcade/adventure game involving a treasure hunt through 120 screens while avoiding a series of hazards. The program features split screen scrolling and secret passageways through walls.

Orpheus in the Underworld retails for £6.95 on the Commodore 64.

Home Computer Line

BRITISH TELECOM (BRADford) have set up the country's first computer information service called 'Home Computer Line'. Callers dialling a Bradford number will hear a three minute tape covering all aspects of computer ownership. The information should initially cover hardware, software, peripherals and computer world news.

For further information contact British Telecom (Bradford), Telephone House, 11 Broadway, Bradford, W. Yorkshire, BD1 1BA; telephone Bradford 392424.

December issue — errata

WE APOLOGISE FOR SOME errors in the article 'The BASIC Facts Pt. 3' (December issue, pp 34-36). They are as follows:

Program 3.1 page 35 — line 130 should read:
130 IF A < 20 GOTO 110

Program 3.2 page 35 — line 130 should read:
130 IF A < 20 GOTO 110

Page 36, column 1 — line 100 should read:
100 IF A <=20 GOTO 100

Program 3.5 page 36, column 3 — line 120 should read:
IF I < >17 then PRINT I

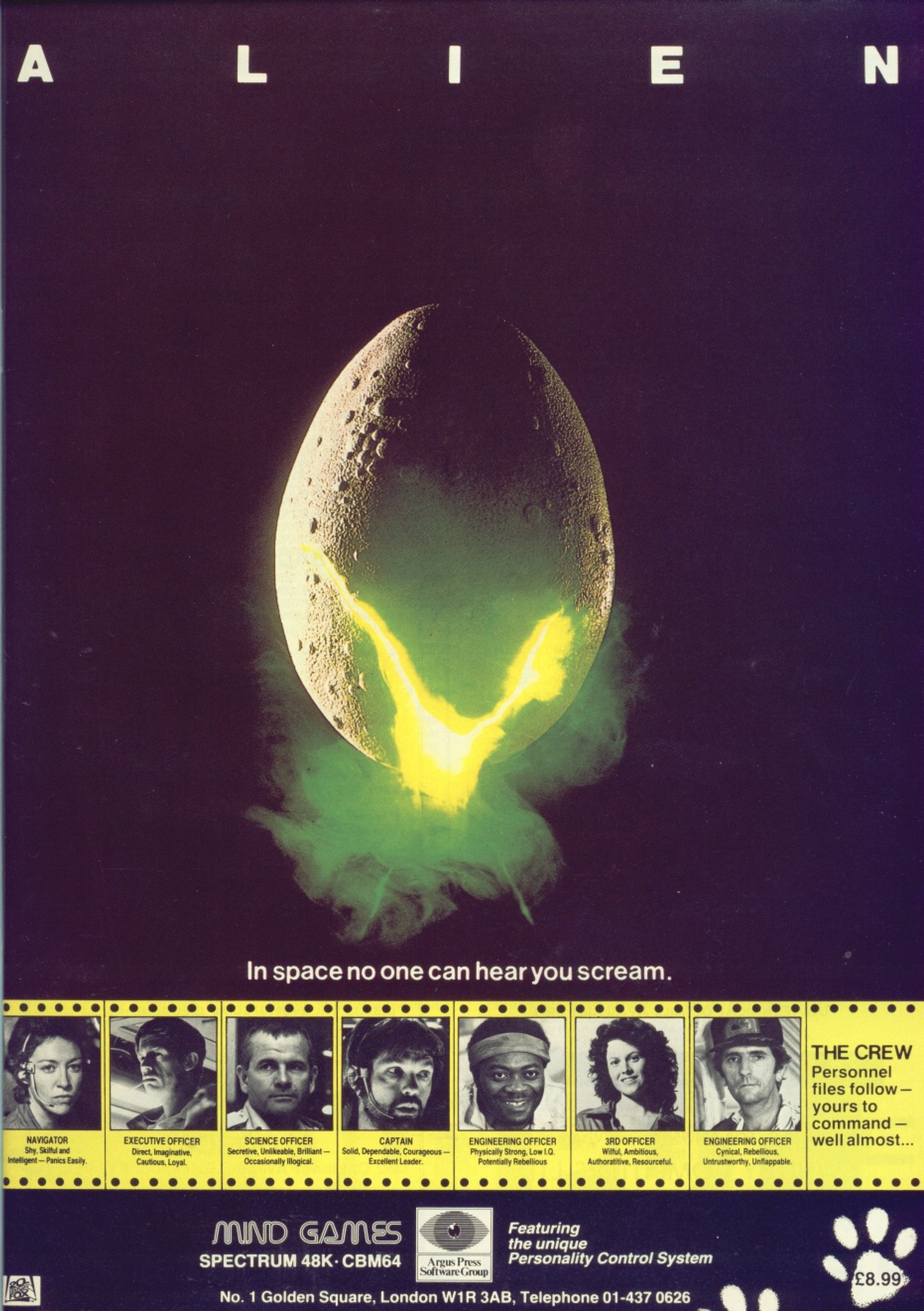
Page 36, column 3, 5 lines from bottom — Line 120 should read:
FOR I = 1 TO 10000

2. We have been bombarded with telephone calls regarding a couple of blurred lines of data statements in the program 'Turbo 64' (pp. 38-39).

Line 255 should read;
255 DATA CC,FF,A9,00,8D,A7,02,A9,01,8D,A8,02,20,44,E5,A9

Line 585 should read:
585 DATA 05,EA,D0,F2,A0,00,A2,04,B1,BB,C4,B7,90,03,A9,20

We apologise for any inconvenience caused to our readers.



ALIEN
In space no one can hear you scream.
NAVIGATOR
Shy, Skilful and Intelligent — Panics Easily.
EXECUTIVE OFFICER
Direct, Imaginative, Cautious, Loyal.
SCIENCE OFFICER
Secretive, Unlikeable, Brilliant — Occasionally Illogical.
CAPTAIN
Solid, Dependable, Courageous — Excellent Leader.
ENGINEERING OFFICER
Physically Strong, Low I.Q. Potentially Rebellious
3RD OFFICER
Wilful, Ambitious, Authoratitive, Resourceful.
ENGINEERING OFFICER
Cynical, Rebellious, Untrustworthy, Unflappable.
THE CREW
Personnel files follow — yours to command — well almost...
MIND GAMES
SPECTRUM 48K · CBM64
Argus Press Software Group
Featuring the unique Personality Control System
£8.99
No. 1 Golden Square, London W1R 3AB, Telephone 01-437 0626

ZX * SPECTRUM

SIMULATOR FOR THE commodore 64 +

ONLY £14 - 95 !

IF YOU HAVE A COMMODORE-64 MICROCOMPUTER, THIS BRILLIANT PROGRAM WILL ALLOW YOU TO USE VIRTUALLY ANY SPECTRUM BASIC PROGRAMS WITHOUT MODIFICATION !

Just think of it ! - your programmes will think you really have a Spectrum ! Because your 64 has a proper keyboard plus RS232, disk-handling, and serial port facilities, we are throwing in a lot of features of ZX-Interface-1 * as well.

WHY YOU SHOULD GET A SPECTRUM SIMULATOR

! You can type in (virtually) all these Spectrum listings from magazines, as well as those for the 64.

! It's an incredibly cheap way to get all the facilities of another computer.

! And if you're upgrading from Spectrum to 64, your old programs don't all become obsolete.

! Spectrum Basic is in many ways easier and more powerful than Commodore Basic. Why not compare them for yourself ?

FULL TECHNICAL DETAILS AVAILABLE FREE

* Trademarks of Sinclair Research Ltd
+ Trademark of Commodore Business Machines Ltd.

ZX-SPECTRUM SIMULATOR

FOR SPEEDY MAIL ORDER SERVICE, SEND CHEQUE/P.O. FOR £14.95 (Access/Visa accepted) to

WHITBY COMPUTERS LTD

7. CHUBB HILL ROAD, WHITBY, N.YORKS, YO21 1JU Tel : 0947-604966

COMMODORE 64

FAST LOADERS TAPE TO DISK

R. B. S. (RAPID BACKUP SYSTEM)

A NEW GENERATION conversion utility, developed from the acclaimed Hypersave-64. R.B.S. converts virtually all your long loading cassette programs to TURBO-LOAD, faster than the CBM disk drive. Multi-part and autorun programs are handled with ease. Converted programs load independently. No additional hardware, no Pokes, no SYS calls, no user knowledge required. Load "The Hobbit" in 120 seconds. Tornado, Lightning load, Fastback now obsolete. R.B.S. will convert more programs than any competing utility. Cassette £7.50p.

HYPERSAVE - 64

Retaining the flexibility which the Programmer demands, Hypersave leaves you in control. 8 additional Basic commands. Save, load, verify up to 48K continuous RAM at HYPERSPEED with full error checking. Converted Programs reload independently. Autorun facility. Hypersave also allows a small number of fast loading programs to be converted to Hyperload. **Cassette £7.50**

BACKUP - DELUXE

The most powerful security backup utility on the market, an unbeatable price. For all standard rate programs. All program sections load and save continuously. Incorporates a sophisticated header reader. Versatile and user friendly. **Cassette £4.50**

NEW

DISKUS I

Introducing the new market leader. This is the simplest to use and most efficient tape to disk transfer utility yet devised. No program rewriting is required and transferred programs will automatically run when loaded. *For single, multi, and autorun programs. *Automatic filename handling. *Efficient use of Disk space. * No user knowledge required. Diskus 1 will handle several proram types which other utilities ignore, including headerless files. A bonus program is included which will transfer a number of fast loading programs to disk. **Cassette £9.99**

PRO - SPRITE

Library, Editor, Animator for single and multicolour sprites. Joystick control. Packed with features. Menu driven. 100% machine code. Complete with your first sprite library. **Cassette £7.50**

DISKUS 1 is **£9.99 SPECIAL DISCOUNT**: any two of the remainder just **£9.99**. All four only **£19.99**. Free secrets! I/O guide with every order.

Fast despatch guaranteed. Cheque/P.O. or S.A.E. to:

DOSOFT

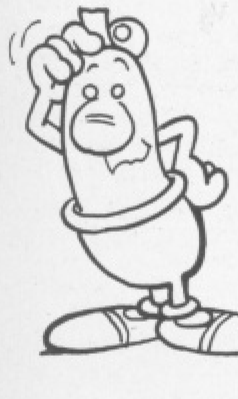
2 Oakmoor Avenue, Blackpool FY2 0EE.

INPUT

If you've got problems then Grahame Davies is here to solve them — if you think we've got problems, let us know!

Dear Sir,
I am writing to ask you how do you program the programmable function keys, F1-F8, on the Commodore 64.
Yours faithfully,
E. Kenworth
Lancashire

We answer,
The function keys on the CBM 64 cannot be programmed from BASIC. If you press one of these keys, it generates a character and you can then test for this character using its ASCII value. They can be "programmed" using a bit of machine code. This patches into the interrupt and tests to see if there is a function key in the first position of the keyboard buffer. If there is, the keyboard buffer is filled with the required command remembering that the buffer is only ten characters long. Program 1 demonstrates this and you can put your own commands into the data statements at the end of the program. Note that a left arrow ← indicates a carriage return.



Program 1

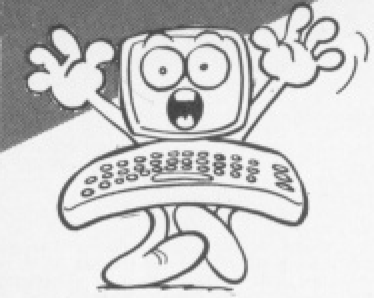
```
10 REM FUNCTION KEY PROG    BY OWEN MANDERFIELD   1/11/84
20 PRINT"LOADING MACHINE CODE":P=12*4096
30 READ A:IFA<>999THENPOKEP,A:P=P+1:GOTO30
40 PRINT"SETTING UP FUNCTION KEYS":P=12*4096+6*16
50 FORI=0TO7:READ A$:PRINTI;A$
60 IFRIGHT$(A$,1)="←"THENA$=LEFT$(A$,LEN(A$)-1)+CHR$(13)
70 A$=A$+CHR$(0)
80 FORJ=1TOLEN(A$):POKEP+I*16+J-1,ASC(MID$(A$,J)):NEXT
90 NEXT:SYS12*4096:END
800 DATA120,169,13,141,20,3,169,192
805 DATA141,21,3,88,96,169,192,72
810 DATA169,26,72,8,72,72,72,76
815 DATA49,234,165,198,205,255,207,208
820 DATA3,76,188,254,170,189,118,2
825 DATA160,0,217,80,192,240,8,200
830 DATA192,8,208,246,76,188,254,152
835 DATA10,10,10,10,168,185,96,192
840 DATA240,9,157,118,2,200,232,224
845 DATA10,208,242,134,198,76,188,254
850 DATA133,137,134,138,135,139,136,140
855 DATA999
870 REM
880 REM  DEFINE YOUR OWN FUNCTION KEY DEFINITIONS HERE
890 REM
900 DATA RUN←,"ꟷLIST",LOAD←,SAVE
920 DATA NEXT,THEN,INPUT,GOTO

READY.
```

Dear Sir,
I am using a Commodore Pet for my project at school, and the school has no handbook for it. I would be grateful if you could answer these questions for me:
a) How do you get auto repeat on all keys on the PET?
b) What are the sound channels and how are they used?
c) What is the PEEK location for reading the keyboard?
Yours faithfully,
Jim Watts
Portsmouth

We answer,

a. You cannot get auto repeat on all of the keys on the PET.
b. Location 59467 controls the sound. POKE this location with sixteen to turn on the sound and POKE it with zero to turn the sound off. If you PRINT CHR$(7) a bell will sound and automatically turn off the sound. Location 231 hold the chime time of the bell. Try POKE231,0:PRINTCHR$(7). This location defaults to sixteen. Location 59466 sets the musical timbre of the sound. POKE this with a number (0-255) to change it. The best values to use are 15, 31 and 51. Location 59464 contains the actual note to be played.
c. Location 151 returns which key is being pressed (255 if no key pressed).
   Location 152 is one when the shift key is being pressed.
   Location 158 holds the number of characters in the keyboard buffer. Thus POKE158,0 will clear the keyboard buffer.
   Location 227 is the length of the keyboard buffer.
   Location 623 contains the characters in the keyboard buffer. If you extend the keyboard buffer by using a command such as POKE 227,40 you will overwrite first cassette buffer which should be fine in nearly all circumstances.



Dear Sir,
I own a Commodore 64 and 'Your Commodore' covers a lot of my interests. I am just getting used to the BASIC language and am taking great interest in your 'BASIC Facts' series.
I would like to see more on graphics, especially 'Fun Graphics', such as how to get 3-D graphics.
Could you tell me how to move a graphic using keys on the keyboard. Also, could you please tell me what a user defined graphic is.
Yours faithfully,
A. Denman,
Barnsley

We answer,
The program over the page demonstrates how to move a sprite about the screen using the keys indicated in the REMS. This may be easily adapted to move any other object.



OUTPUT

Program 2

```
100 POKE 56,60 :CLR               :REM SET TOP OF MEMORY
110 POKE 2040,240                 :REM SET SPRITE 0 AT BLOCK 240 (IE 240*64)
200 P=240*64                      :REM P=START OF BLOCK 240
210 FORI=0TO20:PRINTI"T"          :REM A SPRITO HAS 21 ROWS
220 READ A$
230 FORJ=0TO2:A=0:Z=1             :REM BY 3 BYTES = 24 PIXALS
240 FORK=0TO7                     :REM CONVERT A$ TO BINARY
250 A=A+Z*VAL(MID$(A$,J*8+8-K,1))
260 Z=Z*2:NEXT K
270 POKE P+I*3+J,A                :REM POKE DATA INTO BLOCK
280 NEXTJ:NEXTI
300 V=13*4096                     :REM V=START OF VIC CHIP
310 POKE V+21,1:POKEV+39,7        :REM SET SPRITE 1 ON AND COLOUR TO YELLOW
400 X=100:Y=100                   :REM SET DEFAULT CO-ORDINATES
410 PRINT"SPRITE READY"
500 POKEV,X:POKEV+1,Y             :REM SET X AND Y CO-ORDINATES OF SPRITE 1
505 A=PEEK(197):IFA=64THEN500     :REM ANY KEY HELD DOWN?
510 IFA=62ANDX>0ANDY>0THENX=X-1:Y=Y-1:GOTO500
520 IFA=9ANDY>0THENY=Y-1:GOTO500
530 IFA=14ANDX<255ANDY>0THENX=X+1:Y=Y-1:GOTO500
540 IFA=18ANDX<255THENX=X+1:GOTO500
550 IFA=20ANDX<255ANDY<199THENX=X+1:Y=Y+1:GOTO500
560 IFA=23ANDY<199THENY=Y+1:GOTO500
570 IFA=12ANDX>0ANDY<199THENX=X-1:Y=Y+1:GOTO500
580 IFA=10ANDX>0THENX=X-1:GOTO500
590 GOTO500
600 REM
610 REM  THE ABOVE LINES CHECK FOR KEYS BEING HELD DOWN  AND
620 REM  ADJUST X AND Y ACORDINGLY
630 REM  THE KEYS ARE :-
640 REM
650 REM"   Q   W   E
660 REM"    \  |  /
670 REM"     \ | /
680 REM"
690 REM"   A—  *  —D
700 REM"
710 REM"     / | \
720 REM"    /  |  \
730 REM"   Z   X   C
740 REM"
800 REM DATA FOR SPRITE 0
1000 DATA "111111111111111111111111"
1010 DATA "11                    11"
1020 DATA "11                    11"
1030 DATA "11            111     11"
1040 DATA "11            111     11"
1050 DATA "11                    11"
1060 DATA "11        11          11"
1070 DATA "11        11          11"
1080 DATA "11        11          11"
1090 DATA "11        11         111"
1100 DATA "111111111111        1 11"
1110 DATA "11        11       1  11"
1120 DATA "11        11      1  111"
1130 DATA "11        11     1  1 11"
1140 DATA "11              1  1  11"
1150 DATA "11             1  1   11"
1160 DATA "11            1  1    11"
1170 DATA "11           1  1     11"
1180 DATA "11          1  1      11"
1190 DATA "11         1  1       11"
1200 DATA "111111111111111111111111"

READY.
```

A User Defined Graphic (UDG) is a character specially designed by a programmer (i.e. you!). The CBM 64 has 256 pre-defined characters and a vector pointing at these. They are stored in ROM. You can change the vector to point elsewhere in RAM and define your own characters. On the 64 you have to redefine all of the characters so the best thing to do is to copy all of the character information out of ROM and then redefine as many as you require.

Dear Sir,
I am strongly thinking of purchasing a disc drive for my Commodore 64. But, I have read in different magazines that the Commodore 1541 disc drive is a little bit sluggish. Please could you tell me whether there are any other disc drives which are compatible with the Commodore 64.
Yours faithfully,
Graeme Lee
West Sussex

We answer,
There are no other disc drives that may be directly bolted on to the CBM 64. This is due to the specialised serial port used by Commodore. The speed of the 1541 is mainly limited by the speed of this cable. There is at least one add on extra that you can buy that will considerably speed up the 1541 but might have other memory or buffer limitations. Alternatively you can buy an IEEE cartridge (which should plug onto the user port) and this will open up the whole world of IEEE devices including the Commodore single and twin disc versions as well as hard discs, ten pen plotters and so on. A cartridge like this may cost around £50.

Dear Sir,
Two weeks ago I bought a Commodore VIC 20 computer from a local shop. The computer was reduced in price from £129.95 to £99.95. Not knowing very much about computers, but interested enough in buying one for my own education as well as my children's, I thought this offer represented a good buy. Unfortunately, having bought the item I now find that software availability is being run down for this particular machine. Furthermore, I am informed that the supply of 3-16K RAM may prove difficult as shops are running down stocks in order to make way for the new Commodore 16.

At the time, I was not made aware of the forthcoming change in machines and I am anxious to find out if the new Commodore 16 software can be used on the VIC 20, or if an expansion unit can be obtained to bring it into line with the '16'. If so, where can I purchase such a unit?

Having read your first issue of the magazine, I am delighted that a series on VIC Games Programming is available in an easy to understand format. However, in view of the difficulties described above, could you let me know where I can get games such as 'Multi Tron', described on page 35 of your October issue, and also whether my computer can be updated or not.
Yours faithfully,
P J Lycett,
Stafford

We answer,
Commodore 16 software will not run on the VIC 20. As this machine is being phased out, stocks of games will undoubtedly dwindle although all the current owners will still want to buy new games. There are still plenty of shops stocking a large range of excellent games. If you have trouble getting software or RAM expansion boards then there are certain to be adverts in your local press from people wanting to sell their second hand equipment. The VIC 20 cannot be upgraded to become a Commodore 16.

Sumlock who released MultiTron are at Royal London House, 198 Deansgate, Manchester M3 3NE.

Dear Sir,
I have a VIC 20 and a Zero Electronics 64K RAM card. I want to get into, and use, the 'other' 32K since I understand enough to wreak 32K out of the card. The October issue of Your Commodore said "There is indeed 64K of RAM..." (Mastering Machine Code, p.7): although I have a VIC 20, have I some 'fiddling around' to do with my card?

Hex doesn't vex but all helpers I turn to don't speak English! Can you assist?
Yours faithfully,
B.E. Rivett,
Devon

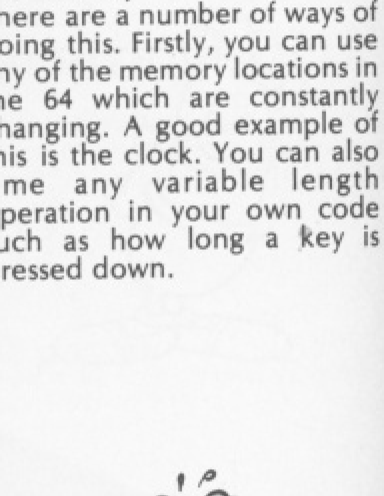
We answer,
The article was referring to the CBM 64 when it said "There is indeed 64K of RAM...". I do not know anything specific about this particular RAM card but there are some guidelines I can give. There must be a manual supplied with it telling you how to use it. If this manual is unclear then try getting in contact with the manufacturers. The principle will probably be one of paging banks of RAM in and out. This means that you use bank one which contains 32K, then you page this out and page bank two in and access the second 32K. To access the first bank, you have to page it back in again. Therefore you can look at either bank but not both at the same time.

Dear Sir,
I have just purchased a Commodore 64. Needless to say I am lost! The hand book tells the minimum. Could you advise me on a better book (or books) where at least the graphic symbols are listed.
Yours faithfully,
T. Davis
Blackburn

We answer,
The best all round guide giving simple and advanced techniques, information and data charts is the Programmers Reference Guide which costs about £15 from "all good computer shops".

Dear Sir,
Please can you help me with a simple problem. I am having great difficulties trying to devise a random number generator in a machine code program on my CBM64. Have you got any ideas?

We answer,
There are a number of ways of doing this. Firstly, you can use any of the memory locations in the 64 which are constantly changing. A good example of this is the clock. You can also time any variable length operation in your own code such as how long a key is pressed down.



We've got the TALENT



and our games prove it!

Amazing graphics, fast and furious action, challenging strategy, compelling adventure – this first wave of games from TALENT has got the lot!

Written by professional computer scientists using powerful new programming techniques (which leave machine code standing), these games have pushed home micros to the very limit.

Kalah

ARCHIPELAGO

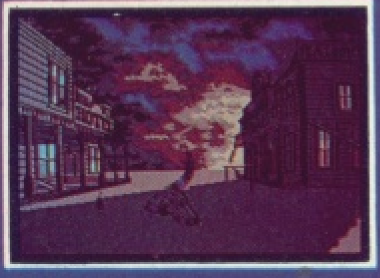












KALAH

One of the oldest games in the world! Originally played by Bedouins using pebbles and holes scooped in the desert sand, this strategy game has absorbed the mind of man for thousands of years. Superb graphics and music, with a choice of levels and depth of search. Easy to learn, fiendishly difficult to beat!

Commodore 64 Cassette £7.95 Disk £9.95

LASER REFLEX

A new angle in alien zapping! Against a brilliant backdrop of stars, you must defend your asteroid base from wave after wave of swooping mutating starships. Your defence – a powerful laser deflected against a giant mirror. How long can you hold out before your defences crumble?

BBC Model B Cassette £7.95
Electron Cassette £7.95

ARCHIPELAGO

Dare you venture into the ancient undersea labyrinths which link the islands of the Archipelago? Fabulous jewels await you there – but the guardians of the treasure mark your every move, waiting to deal a swift and terrible vengeance. How long can you survive?

A fast-moving, action-packed game with brilliant graphics, animation and music.

Commodore 64 Cassette £7.95 Disk £9.95
(joystick needed)

WEST

A compulsive adventure game set in a ghost town in the Wild West. Your task – to flush out a gang of bank robbers and retrieve their loot – with only your trusty steed and your six-shooter to aid you!

Test the speed of your draw, your skill at cards, your powers of survival!

The first of a stunning series of adventure games from Allan Black, played in real-time.

Commodore 64 Cassette £9.95 Disk £12.95
BBC Model (text only) Cassette £7.95
Electron (text only) Cassette £7.95



SOFTWARE FROM SCOTLAND

Talent games should shortly be available from your local software retailer or post off the coupon now.

PANORAMA (H)

An outstanding graphics package for the Commodore 64. Features include: free-hand sketches, circles, ellipses, block move, copy and reverse, magnification choice of cursor, joystick or key control. PLUS texture definition and texture and colour fill. PANORAMA (H) uses the full colour facilities of the Commdore 64.

Commodore 64 Cassette £17.95 Disk £19.95

The programme is menu-driven and comes with a comprehensive illustrative instruction booklet.

TALENT

Curran Building, 101 St James Road
Glasgow G4 0NS Tel 041 552 2128

INSTANT CREDITCARD 24 HOUR ORDERING SERVICE TEL 041 552 2128

TO: TALENT COMPUTER SYSTEMS FREEPOST (no stamp required) GLASGOW G4 0BR
Please send me the following items

COMMODORE 64	Quantity			
PANORAMA (H)	—	Cassette	(£17.95)	☐
PANORAMA (H)	—	Disk	(£19.95)	☐
ARCHIPELAGO	—	Cassette	(£ 7.95)	☐
ARCHIPELAGO	—	Disk	(£ 9.95)	☐
KALAH	—	Cassette	(£ 7.95)	☐
KALAH	—	Disk	(£ 9.95)	☐
WEST	—	Cassette	(£ 9.95)	☐
WEST	—	Disk	(£12.95)	☐

BBC MODEL B	Quantity			
LASER REFLEX	—	Cassette	(£ 7.95)	☐
WEST (text only)	—	Cassette	(£ 7.95)	☐
ELECTRON				
LASER REFLEX	—	Cassette	(£ 7.95)	☐
WEST (text only)	—	Cassette	(£ 7.95)	☐
Total (includes postage and packing)				

I enclose a PO/Cheque to the value of £ ____________

Please debit my ACCESS ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Name (block capitals) ____________

Address ____________

____________ Post Code ____________

Signature ____________ Tel ____________

Please allow up to 28 days for delivery. We shall not bank your remittance until your order has been despatched.
Dealer enquiries welcome. Contact Veronica Colin on 041-552 2128.

Display the time of day on the screen whilst programming with this handy machine code routine from Les Allan.



SCREEN CLOCK

HAVE YOU EVER WONDERED how much time is spent on the family computer when writing or debugging a program? Well here's a chance to find out with a simple screen clock that can be used either as a digital clock to keep an eye on things or as a time elapsed indicator.

Buried away inside the computer is the Complex Interface Adaptor (CIA) chip which performs as well as many other things the Time of Day Clock (TOD). This is a special purpose timer for real time applications. The TOD consists of a 24 hour (AM/PM) clock with $1/10$ sec resolution being organised into 4 separate registers: 10ths of a second, seconds, minutes, and hours. The AM/PM flag is located in



the MSB of the hours register.

A specific sequence of events must be followed for the proper setting of the TOD. TOD is automatically stopped whenever a write is made to the MSB register and will not start again until a write to the 10ths of a second register. This assures that the TOD will always start at any desired time since the entered time is frozen until the final write to the 10th of seconds is made.

Window display

This program creates a window in the top right hand side of the screen to continuously display the time. The interrupt routine is re-routed to a machine code program located high in memory $CF08 so that every time an interrupt occurs the clock can be updated. The only limitation to this program is that the interrupt pointers must not be altered by any other program.

The program works by taking the HEX value of data, converting it to binary coded decimal, carrying out a check to see if it is within the legal range and POKing into memory. Hex data has been used because it keeps the data statement tidy, makes it easier to read and is the working code of the machine. Providing that the final check sum is OK the machine code routine is activated and the display set to enter the correct time of day. Error trapping is again provided to ensure that only the legal range of time can be entered. When the values have been correctly entered, pressing any key will start the clock from that desired time.

Combination changes

The sequence has been arranged to create pleasant colours for the clock, border, screen and cursor as the prime purpose for this routine is to allow a check to be made on time when programming. However, if a change to these combinations is necessary then a few simple pokes is all that is required.

Clock colour	POKE 53113,0-15
Border colour	POKE 53183,0-15
Screen colour	POKE 53188,0-15
Cursor colour	POKE 53193,0-15

Warning!!

As these locations are used by the machine code routine it is essential that no mistakes are made or the program will irrecoverably crash. It is therefore essential that a save is made before funning or changing these values.

To turn on the screen clock at any time all that is required is to enter SYS 53000. To turn off the screen clock enter SYS 53200 or Press RUN/STOP and RESTORE keys simultaneously.

Further information on the TOD can be found from the Programmers Reference Guide Appendix M on page 431.

Program Listing

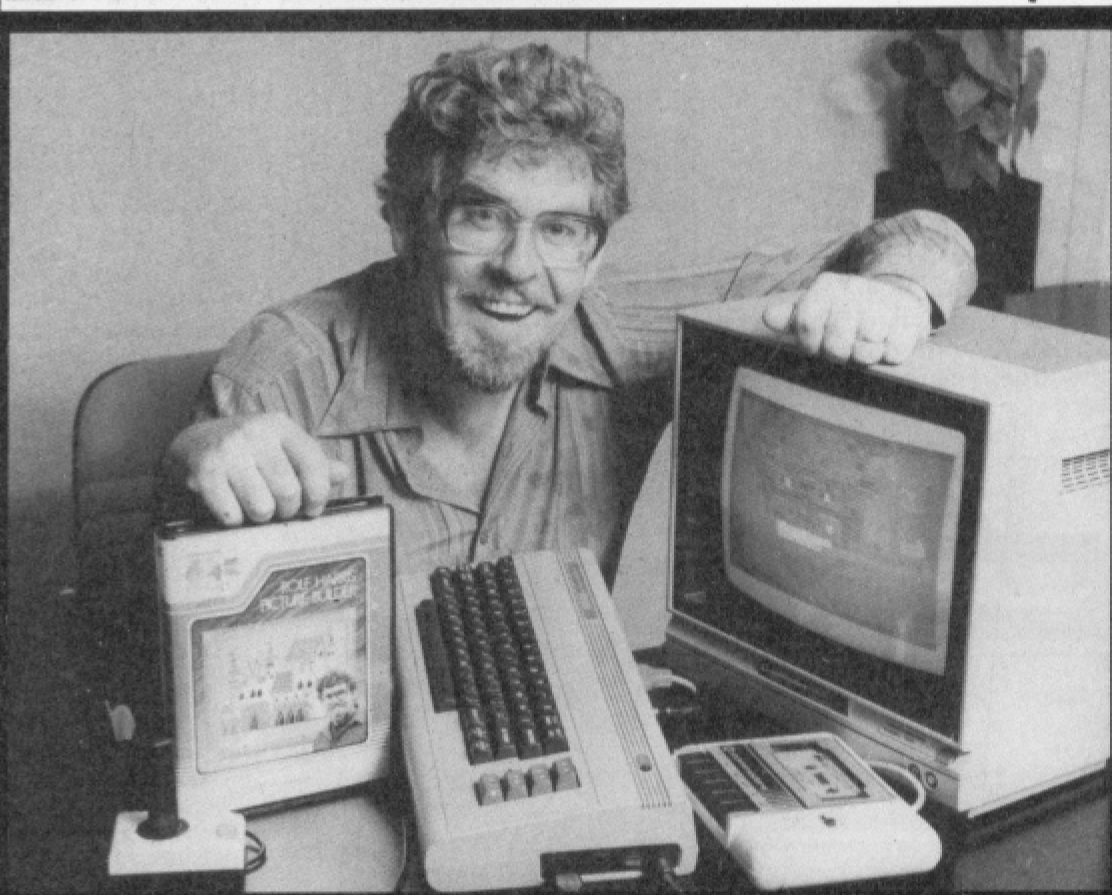
```
READY.

1 :
2 REM ########## SET UP M/C ROUTINE FOR CLOCK ##########
3 :
4 POKE53280,0:POKE53281,0
5 PRINTCHR$(147)CHR$(5)
6 PRINT:PRINTTAB(4)"SETTING UP M/C ROUTINE FOR CLOCK"
7 PRINT:PRINT:PRINT:PRINT
8 PRINTTAB(4)"CLOCK  COLOUR    POKE 53113,0-15"
9 PRINT
10 PRINTTAB(4)"BORDER COLOUR    POKE 53183,0-15"
11 PRINT
12 PRINTTAB(4)"SCREEN COLOUR    POKE 53188,0-15"
13 PRINT
14 PRINTTAB(4)"CURSOR COLOUR    POKE 53193,0-15"
15 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
16 PRINTTAB(9)"CLOCK  ON    SYS53000"
17 PRINT
18 PRINTTAB(9)"CLOCK OFF    SYS53200"
19 :
20 REM ################### HEX LOADER ###################
21 :
22 INC=0:SUM=0
23 READDA$:IFDA$="END"THEN33
24 IFLEN(DA$)<>2ANDDA$<>"END"THEN34
25 H=ASC(LEFT$(DA$,1)):H1=(H-48)*16:IFH>57THENH1=(H-55)*16
26 H=ASC(RIGHT$(DA$,1)):H2=(H-48):IFH>57THENH2=(H-55)
27 BCD=H1+H2:IFBCD<0ORBCD>255THEN34
28 POKE53000+INC,BCD:INC=INC+1:SUM=SUM+BCD
29 GOTO23
30 :
31 REM ############### ERROR TRAP ROUTINE ###############
32 :
33 IFINC=248ANDSUM=28924THEN38
34 PRINTCHR$(147)"ERROR IN DATA STATEMENT !!":STOP
35 :
36 REM ##############  SET UP TIME OF DAY ###############
37 :
38 SYS53000:POKE56335,PEEK(56335)AND127
39 BK$="":FORBK=1TO14:BK$=BK$+CHR$(157):NEXT
40 SP$="                "+BK$
41 PRINTTAB(9)"SETTING UP"
42 PRINTCHR$(17)CHR$(17)TAB(7)"AM PM   "SP$;:INPUTAM$
43 IFAM$="AM"ORAM$="PM"THEN45
44 PRINTCHR$(145)CHR$(145)CHR$(145)CHR$(145):GOTO42
45 A=128:IFLEFT$(AM$,1)="A"THENA=0:
46 PRINTCHR$(17)TAB(7)"HOUR     "SP$;:INPUTHR$
47 IFLEN(HR$)<>2THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO46
48 T=16*VAL(LEFT$(HR$,1))+VAL(RIGHT$(HR$,1))
49 IFT>18THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO46
50 POKE56331,A+T
51 PRINTCHR$(17)TAB(7)"MINUTES "SP$;:INPUTMI$
52 IFLEN(MI$)<>2THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO51
53 T=16*VAL(LEFT$(MI$,1))+VAL(RIGHT$(MI$,1))
54 IFT>89THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO51
55 POKE56330,T
56 PRINTCHR$(17)TAB(7)"SECONDS "SP$;:INPUTSE$
57 IFLEN(SE$)<>2THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO56
58 T=16*VAL(LEFT$(SE$,1))+VAL(RIGHT$(SE$,1))
59 IFT>89THENPRINTCHR$(145)CHR$(145)CHR$(145):GOTO56
60 POKE56329,T
61 PRINTCHR$(17)CHR$(17)CHR$(17)TAB(6)"ANY KEY TO START"
62 GETKEY$:IFKEY$=""THEN62
63 POKE56328,0
64 SYS53200:REM SWITCH OFF CLOCK
65 SYS53000:REM SWITCH ON  CLOCK
66 END
67 :
68 REM ############ HEX DATA FOR M/C ROUTINE ############
69 :
70 DATA A9,EE,8D,98,CF,78,AD,14,03,8D,26,CF,A9,8E,8D,14
71 DATA 03,AD,15,03,8D,27,CF,A9,CF,8D,15,03,58,60,31,EA
72 DATA 78,AD,26,CF,8D,14,03,AD,27,CF,8D,15,03,58,60,AD
73 DATA 18,D0,29,F0,4A,4A,85,FE,A9,1C,85,FD,A0,00,AD,0B
74 DATA DC,48,29,7F,A2,8A,20,82,CF,AD,0A,DC,20,82,CF,AD
75 DATA 09,DC,A2,AE,20,82,CF,AD,08,DC,20,93,CF,68,10,03
76 DATA A9,90,2C,A9,81,20,97,CF,A9,8D,91,FD,A9,D8,85,FE
77 DATA A9,0F,91,FD,88,10,F8,6C,26,CF,48,20,8F,CF,68,20
78 DATA 93,CF,8A,20,97,CF,60,4A,4A,4A,4A,29,0F,09,80,91
79 DATA FD,C8,60,EE,EE,EE,A9,81,8D,0E,DC,AD,98,CF,C9,EE
80 DATA D0,14,EE,98,CF,20,44,E5,A2,00,BD,E3,CF,F0,07,20
81 DATA D2,FF,E8,4C,B2,CF,A9,0F,8D,20,D0,A9,0C,8D,21,D0
82 DATA A9,01,8D,86,02,4C,37,CF,20,44,E5,A9,0E,8D,20,D0
83 DATA 8D,86,02,A9,06,8D,21,D0,4C,28,CF,05,20,20,20,20
84 DATA 20,20,12,20,43,42,4D,20,36,34,20,20,20,43,4C,4F
85 DATA 43,48,20,11,11,92,0D,00
86 DATA END
87 :
88 :
89 ##################################################
90 #                                                #
91 #            SCREEN CLOCK FOR  CBM 64            #
92 #                                                #
93 #      OTHER AUTHORS  NOTE THAT NO COMMODORE     #
94 #     SYMBOLS MAKE FOR EASIER PROGRAM READING    #
95 #                                                #
96 #    LES ALLAN                   21ST SEPT 1984  #
97 #                                                #
98 ##################################################
READY.
```

GRAPHIC SOLUTI
SOLUTIONS GRA

Phil South offers all budding artists a guide to graphic software on the Commodore 64.

THE COMMODORE 64 IS A VERY sophisticated computer; its sound and graphics capabilities make it one of the best devices available to the home user. But, why, oh why was the machine not finished with a state of the art BASIC? This, it seems was an over-sight, in an effort on the part of Commodore to get a 64K computer "out the door" earlier, in case they were beaten to the punch by Japanese Cowboys or English whizz-kids. Specification is everything these days though, so we are stuck with superb facilities that cannot be easily accessed. Or can they? For graphics, we can purchase software solutions to our hardware problems in the form of graphics software.



Graphics Software

Let me demonstrate how a problem to a novice can be easily overcome with software. To draw a line on a computer you have to (a) PLOT an invisible point and (b) DRAW from that point to another point, and for both of those points you need precise co-ordinates like so:

```
10 PLOT 100,100
20 DRAW 50,5
```

This means you've instructed the computer to go from the bottom left of the screen (0,0) to 100 pixels across to the right and 100 pixels upwards (100,100) without making any marks on the screen (PLOT). Then from 100,100 drawing a line 50 pixels up, producing a diagonal line from 100,100 to 150,150. I don't know about you, but this kind of thing gives me a headache after the 400th line of my drawing.

The same problem on the same 64 becomes simple when you boot up your graphics program, like PAINTPIC from Kuma. You can select the "line" option, move the cursor with your joystick or keys to the start position, mark it and move to the finish position. The second method is more accurate and less work, especially in complex drawings where you don't want to have to work out the exact co-ordinates of single pixels.

There are essentially three types of graphics software available to the 64 user: Draw or painting programs, languages, and Games Designers or Sprite Editors.

Draw programs

With Draw programs, or "painting" programs as they are sometimes known, the process of producing graphics screens or titles for your programs is reduced to merely "drawing" your design directly onto the screen, a bit like the old Etch-a-sketch gadgets. They are however a bit more sophisticated than that, as you have the facility to colour individual pixels or lines, or even whole blocks of the screen at will.

There are quite a few draw packages available now, DOODLE by Quicksilva, for instance. This is a disc based drawing system, and as such, is quite advanced (if a teeny bit slow.) You can perform a great many juggling acts with a drawing; copying a section to "stamp" all over the screen, or use as a brushpoint, "box", "line" and "circle" modes, and something called ZOOM, which enlarges an area of the screen, to allow you to draw with greater precision.

Another two similar packages are the cassette based systems, PAINTPIC by Kuma and PANORAMA by Talent. Of the two I find PAINTPIC the better, inspite of the fact it's less user-friendly that PANORAMA, in part because PANORAMA has some really annoying sounds in it. Why do you need sound in a drawing program? I find myself asking the same question. The cursor stutters across the screen to a kind of synthetic bell sound, which on auto-repeat clatters like an empty machine-gun. In the end I had to turn the sound off to concentrate. Not only that, but if you make any mistake like entering an option which is not available, you get the most horrendous noise. I don't know what it's meant to represent, but to me it sounds like a metallic squirrel being cut in half by a circular saw in an echo chamber! (Screeekra-nnnnnnnnnnnnggsssshhhhhhhhhh!!!)



PAINTPIC is a far quieter piece of kit. It is fast, accurate and colourful, making full use of the Commodore's capacity for multicoloured drawings. It's a shame the manual is so difficult to understand.

The last piece I looked at was something called ROLF HARRIS' PICTURE BUILDER. This is a kind of computer aided nursery block program from the originators of the machine itself. You can use the character blocks on the machines keyboard to build up pictures of low-res or character graphics. Simple stuff, you may say, but computer education must begin somewhere, so why not with toddlers?

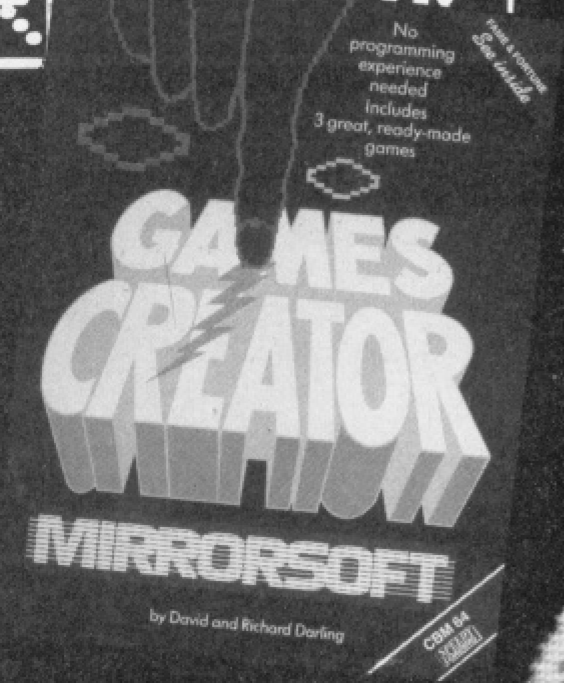
GRAPHIC SOLUTIONS

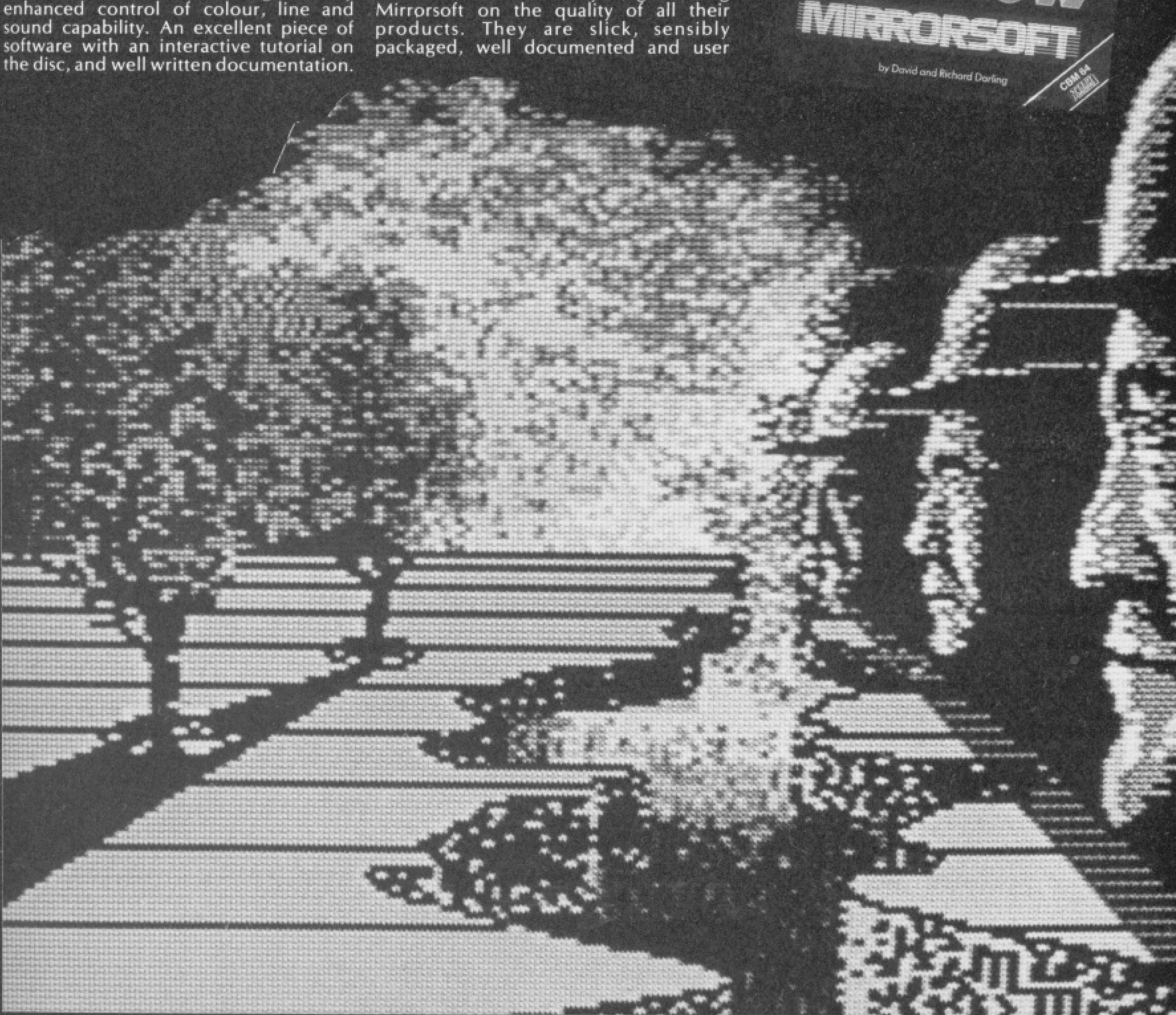
Computer Graphics language

The second type of program is by far the most far reaching and useful of the three, as it is the computer graphics language. ULTRABASIC '64 is the best example of this type of program. The disc based ULTRABASIC 64 is an extended BASIC, boosting the (not to put too fine a point on it) prehistoric BASIC supplied with your machine. This means you can reap the benefits of built in TURTLE functions, and generally enhanced control of colour, line and sound capability. An excellent piece of software with an interactive tutorial on the disc, and well written documentation.

Games Designer and Sprite Editors

The final, and most popular area of interest in graphics at the present is Games Designers and Sprite Editors. Games Designers do just that; allow you to create your own games, which you can play or lend to your friends. Such beasts for the 64 are the excellent GAMES CREATOR and GO SPRITE from Mirrorsoft. Let me start by congratulating Mirrorsoft on the quality of all their products. They are slick, sensibly packaged, well documented and user





GRAPHIC SOLUTIONS SOLUTIONS GRAPHIC



friendly. GAMES CREATOR allows the user to define his own sprites, backgrounds, bullets and sound effects, even music, with no difficulty. And they're not just simple effects either! You've got jumping and gravity effects, scrolling backgrounds, animated sprites, full length musical pieces and explosive, not to say good'n'noisy sound effects. This is a very detailed and satisfying games constructor, the only drawback being the need to drive all your new games using the master program. OK, so you can't sell your game and become rich, but you can re-design your own favourite game. As this type of program goes GAMES CREATOR is very good.

GO SPRITE is a bright and racy sprite creation and animation package, with lots of icons and arrows pointing to applicable parts of the screen. It displays the different "frames" of your sprite like frames of film, and animates them while you watch. You alter the sprite either with keyboard controls or a joystick, moving the cursor to the facility you want to use and pressing the fire button. Of this type of program, Go Sprite is the best I've seen.

Two other packages are not so good. GRAPHICS MASTER is OK but nothing new and a bit oddly laid out. It does however have more colour combinations than many other sprite editors I've seen.

GRAPHICS DESIGNER is possibly the most erratic piece of software I've ever seen. It is "User-violent", in that at no time do you know what it is that the program is doing! The flimsy manual is garbled and the on screen prompts evasive. For instance, what would you make of the phrase "F?". I know what I'd say!

Drawing to an end

The applications of these pieces is up to you. It is possible to use them for games, business graphics, animated cartoons or even art for its own sake. I suggest that before you buy any package that you match its facilities with your skill and your needs.

One last point. I'm fed up with people buying draw programs and doing "abstract computer art" or "computer-type" pictures; using an art program like a computer instead of using a computer like an artistic tool. For goodness sake, when you buy a draw program, DRAW! Be creative. Don't just doodle. There aren't nearly enough people in the world who know about computers and the arts, so there's plenty of room for you in the computer art world of the future.

FIRST AID



for your 64

Computers, like people, are fallible. They need the right combination of code and care to perform effectively in the business or the home. And that requires first hand knowledge from you to create a healthy operating environment for your Commodore 64.

Knowledge about machine language, about the lesser known qualities of the 64, about the disc drives, graphics, and about the tricks and tips to keep your 64 on line. That's why First Publishing has now launched in the UK a series of high quality books and software packages to provide a complete health care kit for your 64. Commodore 64 users throughout Europe have already found it a tonic. We think you will, too.

For a brochure on all the Commodore 64 books and software packages available from First Publishing, please fill in the coupon and send to: Amanda York, First Publishing, Unit 20B, Horseshoe Road, Horseshoe Park, Pangbourne, Berkshire.

Name ______________________

Address ______________________

____

Or ring *Amanda York* at (07357) 5244 or Calco Software at 01-546 7256 YCOM/1

1st
FIRST PUBLISHING LTD

Simon Rockman calls up Commodore's new number, Compunet, an on-line database service for the Commodore 64.

CALLING compunet



COMPUNET IS THE BIGGEST THING that has happened to the Commodore 64 since the machine was launched two years ago. When you buy the Commodore communications modem you get a year's free subscription to the new database service. Anyone who is used to dial-up databases will find Compunet a cross between PRESTEL and bulletin boards. PRESTEL is a passive system where only specially authorised people (Information Providers (IP's) are allowed to put up pages; however, Compunet has had a section called "The Jungle".

Here anyone can put up a page of text or a program and charge for it. When another user wants to read that page or to load that program they can buy it. The money is automatically debited from their account with half of it going to the person who provided the information and the other half to Compunet. The maximum price for a page is £999.99.

When dealing with large sums like this the system needs to be secure against the intruders and hackers who could put up a page under their own name and then use a stolen ID and password to read it. Compunet is one of the most secure public systems invented. Every Compunet modem has its own fingerprint, a code number built into it which is specific to your ID. To use a different account you need a different modem. This level of security should open the door to major transactions such as booking holidays and home banking. The system sounds great so lets look at what happens when you go to use it.

Compunet in use

After the small black box arrives from Commodore you rush to the computer and unwrap the package. The manual is a single flimsy sheet of paper, the rest of it is on the system. (I have never understood why people still like using paper directories when they could use the computer they are talking to do the work of looking things up for them). The modem contains an 8K ROM. This holds your secret serial number and some of the routines necessary for a conversion with the Compunet mainframe. The rest of the routines need to be loaded, from tape, disc or from the mainframe itself. It is in this last form that the program is first sent to you.

To call up the mainframe you simply press shifted C return; this is the connect command and prompts you with the question "Dial Number?". Depending on where you are in the UK you dial a telephone number given in the leaflet which comes with the modem. The modem then dials the number out like a normal telephone. When the modem finishes dialling it prompts you with the message, "Connecting" and then the border changes colour as it connects. Compunet does not have a mainframe to itself (although this is supposed to change); it shares time on a DEC 10 mainframe owned by ADP computer services, whose telephone line it also uses. While the two computers seem to be doing nothing during logging in they are in fact chatting away, logging you into the correct area of the DEC and establishing who you are (from the code number in your modem).

When these formalities are completed, Compunet asks you for your ID and password. The ID is an eight letter code which is used to identify you as the source of any information which you upload. At the moment the password can only be changed by Compunet; however the facility for the user to change the password will become available and will add an extra level of security.

Once logged in, Compunet does the clever trick of interrogating your '64. It finds out what software you have loaded before you logged in. If you have the latest version of the extra software the system lets you straight through. If you have not got anything soft loaded (loaded from tape or disc) then Compunet sends you the software you need. This is called linking and takes a little while. The process is a little slower than using a 1541 disc drive and so, although disc users may decide to save on the cost of the telephone call, tape users may decide to link every time they log in rather than wait for cassette tape to load. If the extra software has been updated since you last saved it the system will send you an update.

Because the mainframe always knows what system you are using (a Commodore 64) it can send you routines and expect you to be able to run them. It would be possible for you to have a program which shares the work between the mainframe and the micro. This would speed up any



games which involved player to player interaction. The mainframe could just send a code and the software in the micro would translate it into a sequence of movements. This is looking to the future; there is a fair bit on Compunet as it is.

The Jungle is fine for posting announcements, but you may want to send a private message to a few friends. For this there is the Courier service. This allows you to send the same information to up to five people. When one of them logs in a small letter box appears on the title page to tell them that there is mail waiting. The first page of the letter shows the recipient who else has had the same letter. No charge has yet been announced but this is likely to be an inexpensive service.

Fun and games

Besides all the boring betting, home banking and estate agent services which make the database profitable for the owners, there is Compucat, a database of rumour and gossip which is run by a computer freak who seems to know some very important people.

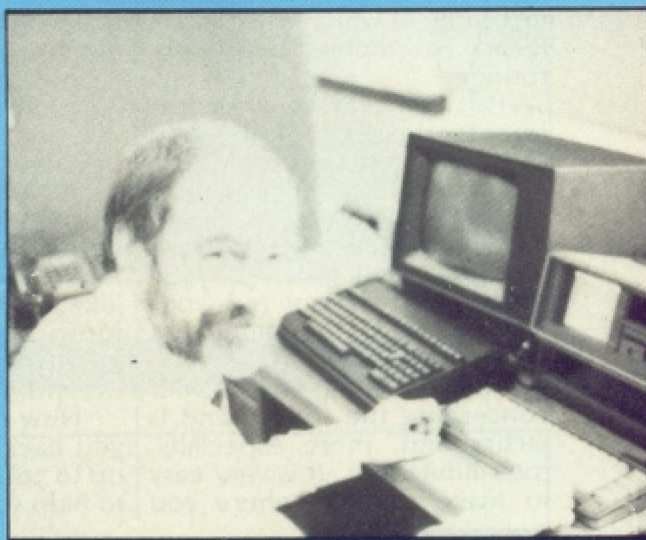
Compucat is not to be confused with Comp-U-Card, a discount ordering scheme which allows you to buy over the modem with a reasonable discount. Because Comp-U-Card requires a scrolling screen the system has to send you some special software, called a link, which supplements the Compunet software.

A similar link is used for the systems adventure game. This is MUD: standing for Multi-User Dungeon, it is the best adventure game written. The vocabulary is much better than that of the Hobbit and the lexical analysis allows for commands like "Get it" and "Kill Him". There are three types of objects: items like the crown and swords which can be carried, mobiles — computer generated assailants which can be fought and may be carrying treasure which can be added to your haul and, thirdly, real people. Other players enter the same fantasy land: you can follow, talk to, fight with or even kiss the other players. MUD is huge; some people get really addicted to it, living as an enchanter or Necromancer is much more fun than being a computer programmer. There are so many facets to MUD that it is impossible to cover them in any one article; the best way to find out about it is to play it. Good luck.

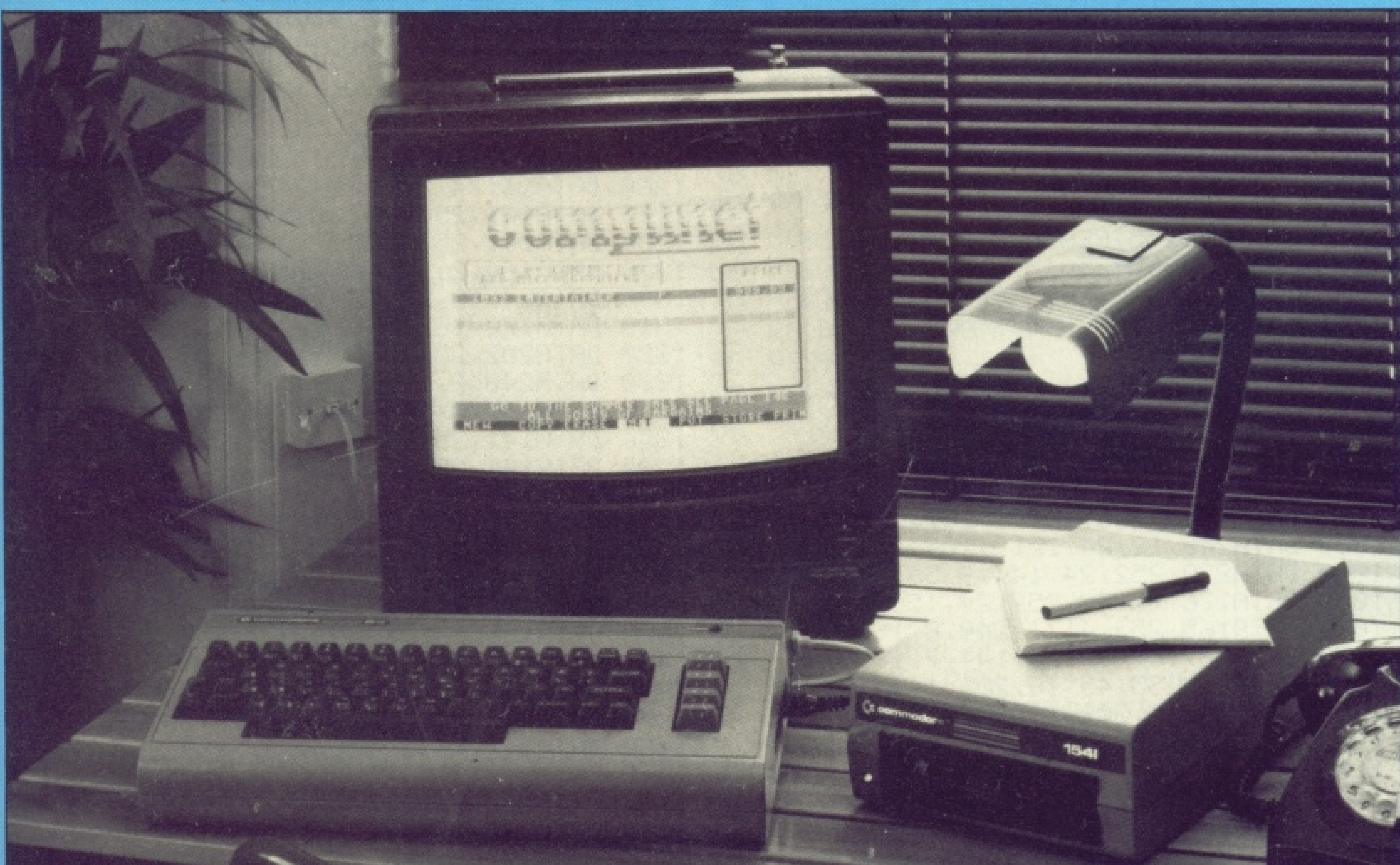
Thor the Wizard has just disappeared in a puff of smoke.----★



Jane Firbank, Editor of Compunet



Nick Green, Compunet's ideas man.



If you need some help in creating your own games, follow this new series from David Rees.

GAMESMANSHIP

IN THE FOLLOWING SERIES, I hope to be able to give you an insight into creating your own fast, impressive and well packaged action games. The series contains four parts, the first three of which will contain ideas, examples and subroutines.

Part one deals with the foreground, probably the most important part of any action game. This emcompasses the objects which are to move, acting out (1) instructions from you (the good guy); (2) your indirect instructions (eg. your laser bolt, or (3) the computer's logic (the bad guys).

The first thing to consider concerning the foreground is tactics and, more especially, their limitations. It is very easy to make a game where you wipe out the enemy in one fell swoop, but this would not be challenging enough. Thus, a game has to be difficult to make it more fun. There are two ways to make a game more challenging: you can make the enemy more intelligent or vicious, or you can limit your own movement/weaponry. The second choice is better in this case as it speeds up the game: Intelligence, or enemy weaponary, takes up plenty of processing time and would turn a BASIC game into a slow joke. Once this foreground format is decided, you can start on the background (which part two will cover).

Now the format is decided, you have to implement it. It is up to you how you do this, but to help speed things up, I have included some machine code subroutines. An explanation of each one follows. Note that the first listing gives a general way to POKE the machine code to memory. You can put as many routines as you want into memory, in any order, and at any sensible location with the following limitations: each routine is no more than 256 bytes long so it is easier if all the routines are spaced this distance apart; the code is used by SYS 'start of the code subroutine'; bytes used in locations 52992-53247 must be reset before using the routines.

Routine one: on using this routine via SYS (routine start), every sprite that you want to move will be moved by a predetermined amount in X and/or Y directions. The routine is very quick, as you can POKE in values before the main part of the game starts, as well as during it. POKE registers as below:

POKE 52992+(sprite no.)	x2, X increment
POKE 52993+(sprite no.)	x2, X add(0) or subtract (1)
POKE 53008+(sprite no.)	x2, Y increment
POKE 53009+(sprite no.)	x2, Y add(0) or subtract (1)

Routine two: this measures whether a collision has occurred between up to eight pairs of sprites that have been defined beforehand. Sprite

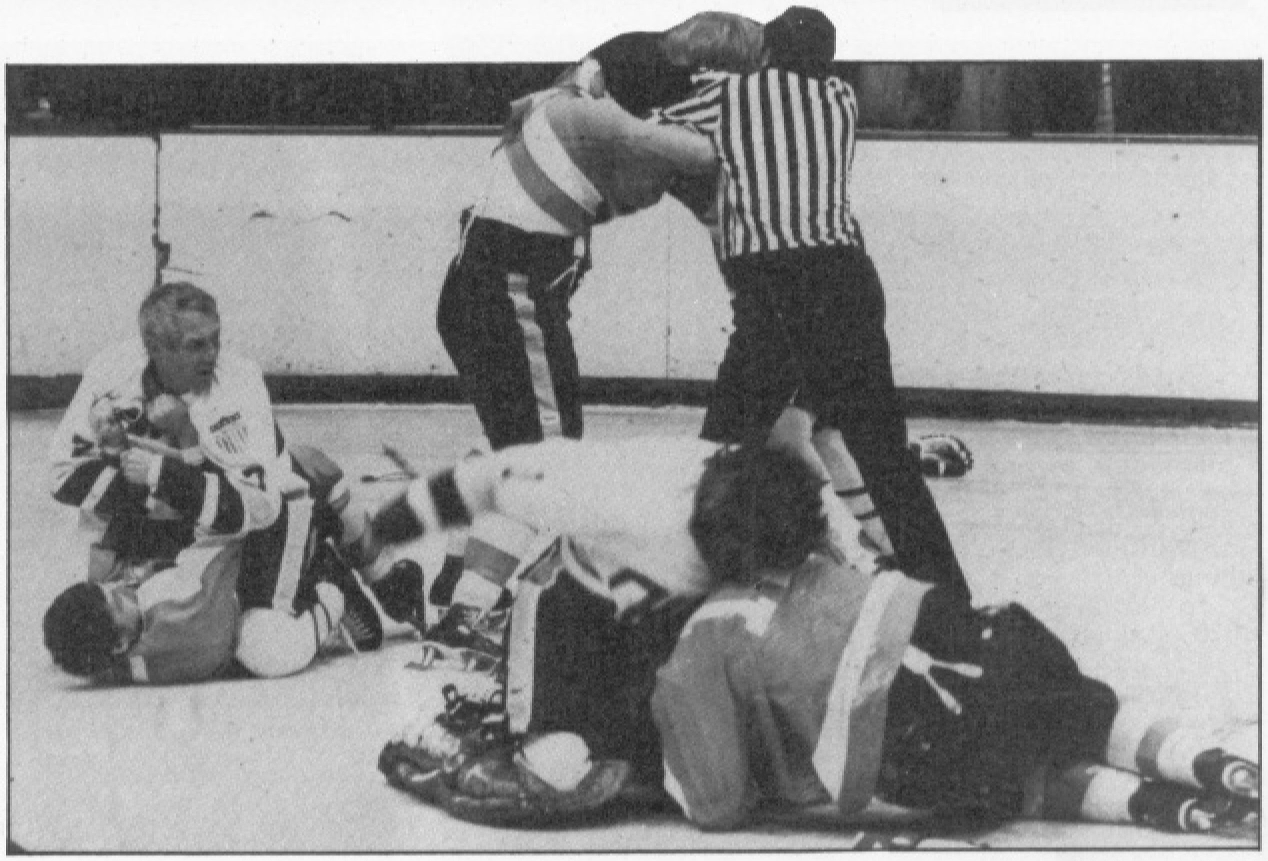
Program Listing 1

```
1 REM********************
2 REM*GAME MACHINE CODE*
3 REM*SUBROUTINES PRT 1*
4 REM*        BY        *
5 REM*   DAVID REES     *
6 REM********************
10 A(1)=13139:A(2)=27770
500 FORN=1TO6:T=0
510 FORM=0TO255:READA:IFA<0THEN530
520 POKE48896+N*256+M,A:T=T+A:NEXT
530 PRINT"■TOTAL FOR ROUTINE"N" IS:"T
540 PRINT"IT SHOULD BE:"A(N)
550 IFH=-2THENEND
560 NEXTN
998 REM* ROUTINE 1 *
999 REM*SPRITE MOVE*
1000 DATA169,1,133,251,160,,173,16,208
1010 DATA37,251,240,2,169,1,133,252
1020 DATA185,,208,190,1,207,224,0
1030 DATA240,12,24,121,,207,144,3,24
1040 DATA230,252,24,144,9,56,249,,207
1050 DATA176,2,198,252,24,153,,208
1060 DATA165,252,41,1,240,8,165,251
1070 DATA13,16,208,24,144,7,165,251
1080 DATA73,255,45,16,208,141,16,208
1090 DATA185,1,208,190,17,207,224,0
1100 DATA240,7,24,121,16,207,24,144,4
1110 DATA56,249,16,207,24,153,1,208
1120 DATA6,251,200,200,192,16,208,154
1130 DATA96,-1
1198 REM*       ROUTINE 2        *
1199 REM*SPRITE/SPRITE COLLISION*
1200 DATA173,32,207,41,7,141,32,207
1210 DATA160,,132,2,185,33,207,201,8
1220 DATA240,111,24,10,24,168,24,74,24
1230 DATA170,169,1,224,,240,6,24,10,24
1240 DATA202,208,250,45,16,208
1250 DATA240,2,169,128,133,251,185,,208
1260 DATA24,74,24,101,251,24
1270 DATA141,250,207,185,1,208,24,74,24
1280 DATA141,251,207
1290 DATA164,2,185,41,207
1300 DATA24,10,24,168,24,74,24
1310 DATA170,169,1,224,,240,6,24,10,24
1320 DATA202,208,250,45,16,208
1330 DATA240,2,169,128,133,251,185,,208
1340 DATA24,74,24,101,251,24
1350 DATA141,252,207,185,1,208,24,74,24
1360 DATA141,253,207
1370 DATA24,144,6,24,144,136,24,144,77
1380 DATA173,250,207,56,237,252,207
1390 DATA176,2,73,255,24,172,32,207
1400 DATA192,,240,5,74,24,136,208,251
1410 DATA201,,240,5,169,,24,144,2
1420 DATA169,1,133,253,173,251,207
1430 DATA56,237,253,207,176,2,73,255
1440 DATA24,172,32,207,192,,240,5,74
1450 DATA24,236,208,251,201,,240,5,169,
1460 DATA24,144,2,169,1,37,253
1465 DATA24,144,2,169,,164,2
1470 DATA153,49,207,200,192,8,208,161
1480 DATA160,,169,,24,121,49,207,24,200
1490 DATA192,8,208,246,96,-2
READY.
```

one of the colliding pair has its number POKEd to 53025+pair no ; sprite two's value is POKEd to 53033+pair no. The result (1= collision) is PEEKed from 53041+pair no, and in addition, by PEEKing 780 directly using this routine, the number of collisions detected will be given. This is a very quick way of telling if a collision has occurred during that go. The last register that you need to know about is 53024, which controls the proximity of the collision (how close the objects' centres have to be for a collision). When POKEing a value here, zero gives a range of two pixels: one gives four pixels, two gives eight pixels etc.

If I have not explained this clearly enough, listing 2 gives examples of how to use each routine.

Next month: Scrolling, sprite/background collisions and gunshots.



Program Listing 2

```
1 REM********************
2 REM* EXAMPLES OF USE *
3 REM*        OF        *
4 REM*GAME MACHINE CODE*
5 REM*SUBROUTINES PRT 1*
6 REM*        BY        *
7 REM*   DAVID REES     *
8 REM********************
9 REM* CODE START BYTES*
10 R1=49152:R2=49152+256
18 REM*SET UP ROUTINE 1 REGISTERS
19 REM*X ADD
20 FOR N=0 TO 7
30 POKE52992+N*2,RND(1)*4
40 NEXT
49 REM* ADD/SUBTRACT
50 FOR N=0 TO 7
60 POKE52993+N*2,INT(RND(1)*2)
70 NEXT
79 REM*Y ADD
80 FOR N=0 TO 7
90 POKE53008+N*2,RND(1)*4
100 NEXT
109 REM* ADD/SUBTRACT
110 FOR N=0 TO 7
120 POKE53009+N*2,INT(RND(1)*2)
130 NEXT
139 REM*SHOW SPRITES
140 V=53248:POKEV+21,255
150 FOR N=0 TO 15
160 POKEV+N,RND(1)*255
170 NEXT
180 POKEV+32,0
190 FOR N=0 TO 7
200 POKEV+39+N,N+8
210 NEXT
298 REM*SET UP ROUTINE 2 REGISTERS
299 REM*PRECISION +/-16 PIXELS
300 POKE53024,3
309 REM*SPRITE NUMBERS*
310 FOR N=0 TO 7
320 POKE53025+N,8
330 NEXT
340 POKE53025,0
350 POKE53033,1
360 POKE53025+1,1
370 POKE53033+1,2
479 REM*[CLR HOME]
480 PRINT"♥"
490 POKE780,0
500 SYSR1:SYSR2:T=T+1:IFT>999THEN700
510 IFPEEK(780)=0THEN500
520 FOR N=0 TO 1
530 X=PEEK(53041+N):IFX<>0THEN560
540 NEXT
550 GOTO700
560 X=V+N+39
570 POKEX,(PEEK(X)+1)AND15
580 POKEX+1,(PEEK(X+1)+1)AND15
590 PRINT"HIT ON REGISTER"N
600 GOTO500
700 RUN
```

Fend off alien and bomb attacks with your powerful laser in this game from F.G. Tout.

LASER TRACK

IN THIS GAME FOR THE Commodore 64, you control a powerful laser that runs on a track. You are, also, under attack from various aliens (and 'things').

To load the game, hit shift/run-stop. Place the joystick in port 2: left = left; right = right; fire button = fire laser.

You score 50 points for each alien you destroy and 100 points for each bomb but, watch those bombs, they will blow up the track. Hit the damaged track or an alien and you lose a life.

You start with 3 lasers and get an extra one every 3rd wave. The game includes 13 waves, 15 levels of difficulty, all 8 sprites, screen scrolling and background music.

Program Listing Part One

```
2 POKE53280,0:POKE53281,0:PRINT"♥"
3 PRINT"████████████                ▓PLEASE WAIT"
5 REM****MUSIC*****
10 FORT=0TO2:FORX=0TO255:READA:IFA=-1THEN40
20 POKE36864+T*256+X,A:NEXTX
40 NEXTT
50 DATA169,146,141,21,3,169,0,141,20,3,96,-1
55 DATA169,234,141,21,3,169,49,141,20,3,96,-1
60 DATA172,0,147,192,17,240,57,206,1,147,173,1,147,201,0,208,44,238,0,147,169
70 DATA0,141,4,212,169,147,141,5,212,169,147,141,6,212,169,33,141,4,212,185,15,1
67
75 DATA185,0,148,141,1,212,185,0,149,141,0,212,185,0,150,141,1,147,76,49,234
80 DATA169,0,141,0,147,141,4,212,24,144,242,-1
110 POKE37632,0:POKE37633,0
120 FORT=0TO85:READHF,LF,DU
130 POKE37888+T,HF:POKE38144+T,LF:POKE38400+T,DU-2
140 NEXT:POKE37380,T+1
150 DATA28,214,24,28,214,18,25,177,14,22,227,12
160 DATA19,63,12,17,37,12,22,227,8,0,0,8,28,214,12
165 DATA28,214,14,25,177,12,22,227,24,19,63,12,17,37,16,22,227,68,0,0,4,28,214,1
2
170 DATA28,214,13,34,75,13,28,214,13,22,227,13,25,177,11,28,214,28,0,0,8
180 DATA25,177,10,25,177,10,22,227,12,21,154,14,19,63,10,17,37,12,22,227,48,0,0,
8
190 DATA32,94,12,34,75,14,43,52,18,38,126,15,34,75,12,32,94,10,25,177,40,0,0,8
200 DATA22,227,14,22,227,14,28,214,14,25,177,12,22,227,16,21,154,14,28,214,32
210 DATA0,0,8
220 DATA22,227,14,25,177,16,28,214,12,25,177,10,22,227,12,19,63,20,17,37,14
230 DATA17,37,12,17,37,12,19,63,16,22,227,34,0,0,8,22,227,32,0,0,8,28,214,34
240 DATA0,0,8,34,75,24
250 DATA38,126,24,0,0,4,43,52,18,34,75,12,32,94,12,25,177,16,0,0,4
260 DATA22,227,10,22,227,10,28,214,16,25,177,12,21,154,12,19,63,24,17,37,30
270 DATA17,37,10,17,37,10,19,63,12,22,227,20,22,227,20,25,177,20,22,227,80
500 REM****SPRITE DATA*****
510 FORI=0TO64*26-1:READA:POKE230*64+I,A:NEXT
520 DATA12,60,48,12,255,32,0,255,32,0,255,32,0,255
530 DATA16,4,60,16,4,0,16,4,20,32,0,85,32,0,85,32,0,85,48,12,85
540 DATA48,12,20,48,12,0,32,0,40,32,0,170,32,0,170,16,4,170,16,4,170
550 DATA16,4,40,32,0,0,32,0
560 DATA0,40,32,0,170,16,12,170,16,12,170,18,12,170,32,0,40,32,0,0,32
570 DATA0,60,48,4,255,48,4,255,48,4,255,32,0,255,32,0,60,32,0,0,16
580 DATA12,20,16,12,85,16,12,85,32,0,85,32,0,85,32,0,20,16,4,0,16,0
590 DATA4,20,16,4,85,32,0,85,32,0,85,32,0,85,48,12,20,48,12,0,48
600 DATA12,40,32,0,170,32,0,170,32,0,170,16,4,170,16,4,40,16,4,0,32
610 DATA0,60,32,0,255,32,0,255,48,12,255,48,12,255,48,12,60,32,0,0,32,0
620 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,60,0,0,255,0,3,215,192,3
630 DATA85,192,3,105,192,3,105,192,3,85,192,3,215,192,0,255,0,0,60,0,0
640 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
650 DATA0,0,0,0,0,0,0,0,0,0,0,0,60,0,0,255,0,3,215,192,15,85,240,13,105,112,13
660 DATA170,112,13,170,112,13,170,112,13,170,112,13,105,112,15,85,240,3
670 DATA215,192,0,255,0,0,60,0,0,0,0,0,0,0,0,0,0,0
680 DATA0,0,0,0,0,0,60,0,0,255,0,3,215,192,15,85,240,61,105,124
690 DATA53,170,92,54,190,156,54,255,156,54,255,156,54,255,156,54,255,156
700 DATA54,190,156,53,170,92,61,105,124,15,85,240,3,215,192,0,255
710 DATA0,0,60,0,0,0,0,0
720 DATA182,216,180,56,35,176,1,194,0,44,153,144,9,35,0,51,132,176,4,118,64
730 DATA2,208,0,1,17,0,1,141,0,0,84,0,0,84,0,0,168,0,0,100,0,0,152,0,0,236,0,0
740 DATA184,0,0,100,0,0,152,0,0,236,0,0,184,0,0
750 DATA187,184,188,52,17,208,2,210,128,52,238,224,14,51,0,51,220,240,0,183,128
760 DATA1,221,0,3,30,0,2,206,0,0,84,0,0,84,0,0,168,0,0,100,0,0,152,0,0,236,0,0
770 DATA184,0,0,100,0,0,152,0,0,236,0,0,184,0,0
800 DATA0,0,0,0,0,0,0,65,0,83,215,197,0,195,5,80,215,0,80,255,5,0,60,5,80,40,0
810 DATA83,170,197,15,150,245,85,150,240,83,170,197
820 DATA0,40,5,80,60,0,80,255,5,0,195,5,80,195,0,3,60
830 DATA197,0,195,0,0,0,0,0
840 DATA0,0,0,0,65,0,3,215,192,80,195,5,80,215,0,0,255,5,80,60,5
850 DATA80,60,0,0,40,5,83,170,197,95,150,240,15,150
860 DATA245,83,170,197,80,40,0,0,60,5,80,255,5,80,195
870 DATA0,0,195,5,83,60,197,0,195,0,0,0,0,0
880 DATA0,65,0,3,215,192,0,195,0,0,215,0,80,255,5,80,60,5,0,60,0
890 DATA80,60,5,80,40,5,3,170,192,95,150,245,95,150,245,3,170,192,80,40,5
900 DATA80,60,5,0,255,0,80,195,5,80,195,5,3,60,192,0,195,0,0,0,0,0
910 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
920 DATA4,0,2,29,32,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
930 DATA0,0
940 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
950 DATA4,0,32,29,2,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
960 DATA0,0
970 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
980 DATA4,0,0,29,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
990 DATA0,0
1000 DATA1,0,16,1,64,80,0,64,64,0,81,64,0,81,64,0,17,0,0,21,0
1010 DATA0,0,0,0,38,0,0,145,128,0,145,128,0,38,0,0,0,0,0,0,0
1020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1030 DATA0,0,0,0,0,0,0,0,0,20,0,5,5,64,84,1,81,80,0,21,0
1040 DATA0,0,0,0,38,0,0,145,128,0,157,128,0,38,0,0,0,0,0,0,0
1050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1060 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,64,64,1,81,80,5,21,20
1070 DATA20,0,5,0,38,0,0,157,128,0,157,128,0,38,0,0,0,0,0,0,0
1080 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1090 DATA0,0,0,0,0,0,0,0,0,0,130,136,0,130,0,10,130,8,0,130,0
2000 DATA0,130,136,0,0,0,0,0,0,0,0,0,0
2010 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2020 DATA0,0,0,0,0,0,0,0,0,4,69,68,4,65,4,5,65,4,4,65,0
2030 DATA4,69,68,0,0,0,0,0,0,0,0,0,0
2040 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2050 DATA0,0,0,0,0,0,0,0,0,12,207,204,12,195,12,15,195,12,12,195,0
2060 DATA12,207,204,0,0,0,0,0,0,0,0,0,0
2070 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3000 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3010 DATA40,0,0,190,0,1,170,64,0,170,0,0,40,0,0,0,0,0
3020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3030 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3040 DATA40,0,0,170,0,5,190,80,0,170,0,0,40,0,0,0,0,0
3050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3060 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3070 DATA40,0,0,170,0,85,170,85,0,190,0,0,40,0,0,0,0,0
3080 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4000 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,34,0,0
4010 DATA0,0,0,136,128,0,0,0,0,34,0,0,0,0,0,0
4020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4030 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4040 DATA0,0,0,42,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4060 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4070 DATA0,0,0,0,0,0,0
4080 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5000 POKE53280,0:POKE53281,0:PRINT"♥"
5002 POKE53281,0:POKE53280,0:PRINT"█████                  INSTRUCTIONS▓"
5005 PRINT"▙ LASER TRACK BY F G TOUT.   YOU CONTROL   ";
5010 PRINT"█ A POWERFUL LASER THAT RUNS ON A TRACK   ";
5020 PRINT"█ AND YOU ARE UNDER ATTACK FROM VARIOUS   ";
5025 PRINT"▙ ALIENS (& THINGS).   JOYSTICK IN PORT 2 ";
5030 PRINT"█ LEFT=LEFT RIGHT=RIGHT F/B TO FIRE LASER";
5035 PRINT"▙ YOU SCORE 50 PTS FOR EACH ALIEN YOU     ";
5040 PRINT"█ DESTROY,AND 100 PTS FOR EACH BOMB,BUT   ";
5045 PRINT"▙ WATCH THOSE BOMBS, THEY WILL BLOW UP    ";
5050 PRINT"█ THE TRACK. HIT THE DAMAGED TRACK AND    ";
5055 PRINT"▙ YOU LOSE A LIFE. HIT AN ALIEN AND YOU   ";
5060 PRINT"█ WILL ALSO LOSE A LIFE. YOU START WITH 3";
5065 PRINT"▙ LASERS,AND GET AN EXTRA ONE EVERY 3RD   ";
5070 PRINT"█ WAVE. THERE ARE 13 WAVES,15 LEVELS OFF ";
5075 PRINT"▙ DIFFICULTY,ALL EIGHT SPRITES,SCREEN     ";
5080 PRINT"█ SCROLLING,AND BACKGROUND MUSIC  (?)     ";
5085 PRINT"▙ PRESS SPACE TO LOAD GAME ":POKE198,0
5090 GETA$:IFA$=""THEN5090
5095 IFA$=" "THENPOKE53280,4:GOTO50000
6000 GOTO5090
50000 POKE198,2:POKE631,13:LOAD
READY.
```

Program Information

Part 1	
10 — 499	Music (interrupt and data)
500 — 4080	Sprite data
5000 -------	Instructions
Part 2	
5 — 499	Download U.D.G.s
30000 — 35119	Sprite data
39999 — 49320	M/Code data (REMmed)
Part 3	
1 — 130	Clr sound and set variables
200 — — —	Define function
500 — 530	Set sprite variables and colour
1000 — — —	Go to title page, clr regs.
4000 — 4099	Main routine
500 — 5099	Check if track damaged
7005 — 7110	Check if bomb hit track
7700 — 8010	Check collision regs
8300 — 8310	Sound
9100 — 9799	Change aliens for waves
9800 — 9960	Cleared wave !!!
10000 — 10199	Screen
20000 — — — —	Laser hit and decrement lives
27000 — 27010	Game over
40000 — 40120	Score, lives, etc.
60000 — 60130	Title page
62000. — 62080	Key reference
63000 — — — —	Game intro........

Variables

HI = high score	LD = level of difficulty
DD = wave counter	V = sprite variable
WA = wave	N = screen position
B1 = m/c address	L = joystick
WE = collision reg	SC = score
L = lives	S1,S2,S3 = sound regs
RR = music counter	EN = enemies

Please type in parts 1,2 and 3 separately and save each one before running the program.

All the machine code routines are REMmed so you may be able to use them in your own routines.

Once the title page appears you will notice that you have the choice of the following:

F1 Increase speed of aliens (LD)
F3 Increase number of aliens per wave (EN)
F5 Music on
F7 Music off
Clr/home set
K Key reference (if you forget)



Finally, you can have as many aliens as you wish: the minimum is 10, the maximum is whatever you decide.

Program Listing. Part Two

```
5 POKE53280,2:POKE53281,2:PRINT"[redacted]                PLEASE WAIT"
10 CLR:CC=0:A=0
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORG=0TO64*8:POKE12288+G,PEEK(53248+G):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
100 READAX:T=T+1:IFAX<>-1THENPOKE12288+G+T,AX:CC=CC+1+AX:GOTO100
101 FORI=0TO60*8STEP8:A=12288+1+I:B=PEEK(A):POKEA,B AND0:NEXT
102 DATA0,0,0,0,0,0
105 DATA1,19,158,55,95,127,127,255,128,208,240,241,226,246,254,255
110 DATA255,51,102,204,255,102,102,255,143,113,156,103,249,142,97,254
115 DATA112,142,99,152,6,113,158,1,0,0,0,3,15,127,127,63
120 DATA0,0,152,252,255,255,255,255,0,0,0,192,240,254,254,252
125 DATA0,0,24,60,24,0,0,0
130 DATA194,113,8,34,225,22,99,255
135 DATA255,255,195,195,195,195,255,255,255,255,195,195,195,195,195,195
140 DATA1,3,7,15,31,63,127,255,255,254,252,248,240,224,192,128
145 DATA81,247,255,255,255,255,255,255,255,255,255,255,255,255,247,81
150 DATA255,255,255,255,255,255,255,255
499 DATA-1
30000 FORI=0TO127:READA:POKE210*64+I,A:CC=CC+1+A:NEXT
30001 DATA0,0,0,0,0,0,0
30002 DATA0,0,0,0,0,0,0
30003 DATA0,0,0,0,0,16,0
30004 DATA0,40,0,0,32,0,0
30005 DATA32,0,0,32,128,10,34
30006 DATA128,2,170,0,0,160,0
30007 DATA0,84,0,1,69,0,37
30008 DATA205,64,60,252,244,175,255
30009 DATA232,160,0,160,0,0,16,0
30011 DATA0,0,0,0,0,0,0
30012 DATA0,0,0,0,0,0,0
30013 DATA0,0,0,0,0,16,0
30014 DATA0,40,0,0,32,0,0
30015 DATA32,0,0,32,128,10,34
30016 DATA128,2,170,0,0,160,0
30017 DATA0,84,0,1,69,0,37
30018 DATA205,96,100,252,240,47,255
30019 DATA224,160,0,160,32,0,32,0
30020 FORI=0TO63:READA:POKE212*64+I,A:CC=CC+1+A:NEXT
30030 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,20,0,0,20,0,0,20,0
30040 DATA0,20,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
30050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
30200 FORI=0TO255:READA:POKE213*64+I,A:CC=CC+1+A:NEXT
30210 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,0,0,204,0,3,35,0,12,168,192,143
30220 DATA87,200,142,170,200,170,170,168,34,34,32,10,170,128,3,155,0,15,3,192
30230 DATA60,0,240,0,0,0,0,0,0,0,0,0,0,0,0,0
30240 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,0,0,204,0,3,35,0,12,168,192,47
30250 DATA87,224,46,170,224,170,170,168,0,136,128,10,170,128,3,188,0,15,3,192,12
30260 DATA0,192,15,3,192,0,0,0,0,0,0,0,0,0,0
30270 DATA0,0,0,0,36,0,0,85,0,0,101,192,1,221,84,38,101,236,25,247,84,23,153
30280 DATA216,109,125,149,153,231,121,93,218,86,86,119,165,84,149,149,29,103,88
30290 DATA45,217,228,6,119,80,1,231,64,0,109,0,0,150,0,0,44,0,0,16,0,0
30300 DATA0,16,0,0,100,0,1,85,64,7,117,144,13,153,80,38,85,236,25,222,85,93,149
30310 DATA217,89,117,151,149,103,121,233,89,87,117,119,109,90,85,149,29
30320 DATA219,120,45,149,148
30330 DATA6,125,80,5,214,208,1,109,64,0,150,0
30350 DATA0,44,0,0,16,0,0
33000 FORI=0TO63:READA:POKE217*64+I,A:CC=CC+1+A:NEXT:REM**BOMB**
33001 DATA0,0,0,0,0,0,0
33002 DATA0,0,2,170,128,2,60
33003 DATA128,2,170,128,0,40,0
33004 DATA0,40,0,0,40,0,0
33005 DATA170,0,2,234,128,2,250
33006 DATA128,2,190,128,2,190,128
33007 DATA2,175,128,0,170,0,0
33008 DATA40,0,0,0,0,0,0
33009 DATA0,0,0,0,0,0,0,0
34000 FORI=0TO191:READA:POKE218*64+I,A:CC=CC+1+A:NEXT
34001 DATA0,0,0,0,0,0,0
34002 DATA0,0,255,255,255,0,20
34003 DATA0,0,40,0,0,170,0
34004 DATA2,170,128,2,150,128,2
34005 DATA190,128,2,85,128,0,85
34006 DATA0,0,40,0,0,170,0
34007 DATA2,195,128,2,0,128,15
34008 DATA195,240,15,195,240,0,0
34009 DATA0,0,0,0,0,0,0,0
34101 DATA0,0,0,0,0,0,0
34102 DATA0,0,11,255,224,0,20
34103 DATA0,0,40,0,0,170,0
34104 DATA2,170,128,2,150,128,2
34105 DATA190,128,2,85,128,0,85
34106 DATA0,0,40,0,0,170,0
34107 DATA2,195,128,2,3,240,15
34108 DATA195,240,15,192,0,0,0
34109 DATA0,0,0,0,0,0,0,0
35001 DATA0,0,0,0,0,0,0
35002 DATA0,0,0,190,0,0,20
35003 DATA0,0,40,0,0,170,0
35004 DATA2,170,128,2,150,128,2
35005 DATA190,128,2,85,128,0,85
35006 DATA0,0,40,0,0,170,0
35007 DATA2,195,128,15,192,128,15
35008 DATA195,240,0,3,240,0,0
35009 DATA0,0,0,0,0,0,0,0
35010 FORI=0TO191:READA:POKE222*64+I,A:CC=CC+1+A:NEXT:REM***PATTEN*****
35015 DATA0,0,0,64,0,4,20,0,80,1,85,0,0,16
35020 DATA0,30,121,228,32,0,12,51,158,72,17,0,196,34
35025 DATA20,140,51,0,72,17,100,196,32,0,140,54,219,72
35030 DATA0,0,4,102,219,100,0,16,0,1,85,0,20,0,80,64,0,4,0,0,0,0
35035 DATA0,0,0,0,0,0,0,0,0,85,85,84,0,16,0,27,109,100,40,0,0
35040 DATA34,218,76,17,0,132,51,36,200,34,12,76,17,216,132,40,0,200,39,158,76
35045 DATA0,0,4,231,158,120,0,16,0,85,85,84,0,0,0,0,0,0,0,0,0
35050 DATA0,0,0,0,0,0,0,0,0,1,85,0,20,16,80,121,231,156,16,0,0
35055 DATA34,121,196,51,0,140,17,56,72,34,4,196,51,100,140,16,0,72,45,102,196
35060 DATA0,0,12,109,102,216,20,16,80,1,85,0,0,0,0,0,0,0,0,0,0,0
35100 FORI=0TO127:READA:POKE225*64+I,A:CC=CC+1:NEXT:REM**SCORPIAN**
35101 DATA0,0,0,0,1,128,0
35102 DATA5,0,0,5,0,32,20
35103 DATA0,136,20,34,130,20,130
35104 DATA0,150,0,32,60,0,136
35105 DATA215,34,130,195,130,0,215
35106 DATA0,2,60,128,0,0,32
35107 DATA32,0,0,128,0,2,160
35108 DATA0,10,128,0,2,32,0
35109 DATA8,0,0,0,0,0,0,0
35111 DATA0,0,0,2,64,0,0
35112 DATA80,0,0,80,0,0,20
35113 DATA0,0,20,0,2,20,128
35114 DATA0,150,32,32,60,0,128
35115 DATA215,2,130,255,130,0,215
```

Program Listing

```
35116 DATA32,34,190,136,136,0,34
35117 DATA136,0,34,2,0,128,0
35118 DATA130,0,0,170,0,0,170
35119 DATA0,0,0,0,0,0,0,0
39999 FORI=0TO111:READA:POKE49152+I,A:CC=CC+1+A:NEXT:REM***SCROLL LEFT # RIGHT*****
40000 DATA 169,0,133,251,169,4,133,252,169,0,133,254,160,0,177,251
40010 DATA 133,253,160,1,177,251,136,145,251,200,200,192,40,208,245,160
40020 DATA 39,165,253,145,251,165,251,105,39,133,251,144,2,230,252,230
40030 DATA 254,166,254,224,23,208,213,96
40040 DATA 169,0,133,251,169,4,133,252,169,0,133,254,160,39,177,251
40050 DATA 133,253,160,38,177,251,200,145,251,136,136,192,255,208,245,160
40060 DATA 0,165,253,145,251,165,251,105,39,133,251,144,2,230,252,230
40070 DATA 254,166,254,224,23,208,213,96
40999 FORI=0TO67:READA:POKE49266+I,A:CC=CC+1+A:NEXT:REM*****ME L&R*****
41000 DATA 173,0,208,201,50,208,7,173,16,208,41,1,208,18,24,238
41010 DATA 0,208,238,0,208,208,9,173,16,208,73,1,141,16,208,24
41020 DATA 96,173,0,208,201,42,208,7,173,16,208,41,1,240,18,56
41030 DATA 206,0,208,206,0,208,208,9,173,16,208,73,1,141,16,208
41040 DATA 24,96,0,0
41050 FORI=0TO73:READA:POKE49990+I,A:CC=CC+1+A:NEXT:REM*****J/S*****
41100 DATA 173,0,220,201,123,208,12,32,56,192,32,147,192,32,181,192,32,173,194
41110 DATA 201,119,208,12,32,0,192,32,114,192,32,181,192,32,136,194
41120 DATA 201,107,208,12,32,56,192,32,147,192,32,181,192,32,173,194
41130 DATA 201,103,208,12,32,0,192,32,114,192,32,181,192,32,136,194
41140 DATA32,192,193,32,36,194,96
41199 FORI=0TO14:READA:POKE49333+I,A:CC=CC+1+A:NEXT:REM*****CHANGE ME*****
42000 DATA 173,248,7,201,211,240,4,238,248,7,96,206,248,7,96
42100 FORI=0TO42:READA:POKE49400+I,A:CC=CC+1+A:NEXT:REM**MISSILE*****
42110 DATA206,3,208,206,3,208,206,3,208,206,3,208,206,3,208,206,3,208,206,3,208
42120 DATA206,3,208,206,3,208,206,3,208,206,3,208,206,3,208,206,3,208,206,3,208
42130 DATA96
43999 FORI=0TO73:READA:POKE49800+I,A:CC=CC+1+A:NEXT:REM
44000 DATA 206,4,208,206,4,208,206,6,208,206,6,208,206,8,208,206
44010 DATA 8,208,206,10,208,206,10,208,206,12,208,206,12,208,206,14
44020 DATA 208,206,14,208,96,238,4,208,238,4,208,238,6,208,238,6
44030 DATA 208,238,8,208,238,8,208,238,10,208,238,10,208,238,12,208
44040 DATA 238,12,208,238,14,208,238,14,208,96
49200 A=0:FORI=0TO27:READA:POKE49600+I,A:CC=CC+1+A:NEXT
49210 DATA162,4,189,1,208,24,105,4,157,1,208,201,245,144,6,173,4,220,157,0,208
49220 DATA232,232,224,16,208,231,96
49300 B=0:FORI=0TO22:READB:POKE49700+I,B:CC=CC+1+B:NEXT
49310 DATA162,3,189,248,7,24,105,1,201,215,144,2,169,213,157,248,7,232,224,8
49320 DATA208,236,96
63000 IFCC<>128525THENPRINT"DATA ERROR":END
63100 REM*****PLEASE SAVE BEFORE RUNNING*****
63999 POKE198,2:POKE631,13:LOAD
READY.
```

Program Listing Part Three

```
1 CLR:HI=0:LD=2:DD=1:V=53248:WA=0:POKE49709,215:POKE49713,213
2 B=49990:N=1984:B1=49400
5 REM***VARS***
10 POKE53272,29:L=56320:WE=53278:SC=0:LI=3:VR=53279
20 FORI=54272TO54296:POKEI,0:NEXT:S1=54290:S2=54291:S3=54287:POKE54296,15
25 GOSUB60000:LI=3:WA=0:DD=1:LE=0:SC=0:FORI=2042TO2047:POKEI,213:NEXT:EL=0
30 POKE49709,215:POKE49713,213
100 A$="●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
110 B$="~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
120 C$="────────────────────────────────────────"
130 D$="─||||||||| |"
199 REM*****DEFF*****
200 DEFFNA(A)=(PEEK(V+30)AND1)
499 REM****SP VAR ETC.*****
500 POKEV+21,253:POKEV,50:POKEV+1,210:POKEV+28,253:POKEV+37,2:POKEV+38,6
510 POKEV+39,7:POKEV+40,6:POKEV+41,3:POKE2040,210:POKE2042,217:POKE2041,212
520 GG=255:FORI=5TO15STEP2:GG=GG-31:POKEV+I,GG:POKEV+I-1,0:NEXT
530 FORI=42TO47:POKEV+I,0:NEXT
999 REM***MAIN ROUTINE*****
1000 GOSUB10000:POKEV+30,0:POKEV+31,0
4000 J=INT(PEEK(V)/8)+N:U=PEEK(L)
4006 IFU<112ANDU>102THENGOSUB7700
4007 IFLE>ENTHENLE=0:DD=DD+1:GOSUB9500
4010 GOSUB40100:GOSUB5000
4020 IFLI<1THENGOSUB27000:GOTO25
4099 GOTO4000
5000 IFPEEK(J)=67THEN5010
5005 GOSUB20000
5010 SYSB:GOSUB7005:POKEV+2,PEEK(V)
5020 IFFNA(A)=1THENGOSUB20000
5099 RETURN
7005 K=INT(PEEK(V+4)/8)+N-1:IFPEEK(V+5)>205ANDPEEK(V+5)<220THENPOKEK,74:GOSUB7100
7006 IFPEEK(V+5)>245THENPOKE2042,217:RETURN
7010 RETURN
7100 POKE2042,215:RR=55:GOSUB8300:RETURN
7110 RETURN
7700 POKEV+21,255:POKES1,128:POKES2,9:POKES3,10:POKES1,129
7705 IFPEEK(V+3)>94THENSYSB1:SYSB1:GOSUB7900:GOSUB8100:GOTO7705
7800 POKEV+21,253:POKEV+3,200:RETURN
7900 JJ=PEEK(VR):IF(JJAND2)=2THENRR=2:GOSUB8300:GOTO7902
7901 RETURN
7902 IFJJ=130THENF1=2047:F2=15:F3=14
7903 IFJJ=6THENF1=2042:F2=5:F3=4:SC=SC+50
7904 IFJJ=10THENF1=2043:F2=7:F3=6
7905 IFJJ=18THENF1=2044:F2=9:F3=8
7906 IFJJ=34THENF1=2045:F2=11:F3=10
7907 IFJJ=66THENF1=2046:F2=13:F3=12
7908 POKEF1,PEEK(49713):POKEV+F2,30:POKEV+F3,INT(RND(1)*255):POKE2042,217
7910 POKEV+21,253:POKEV+30,0:SC=SC+50:LE=LE+1:EL=EL+1
7950 POKEV+40,2:POKE2041,216:POKEV+21,253:POKE2041,212:RETURN
8010 RETURN
8100 T=T+1:IFT>1THENT=0:SYS49990:RETURN
8110 RETURN
9100 POKE49709,215:POKE49713,213:LD=LD+1:POKE49607,LD:RETURN
9500 WA=WA+1:GG=255:FORI=5TO15STEP2:GG=GG-31:POKEV+I,GG:POKEV+I-1,0:NEXT:GOSUB9800
9503 ONDDGOSUB9510,9520,9530,9540,9550,9560,9570,9580,9590,9600,9610,9620,9630
9505 LD=LD+1:POKE49607,LD:RETURN
9510 FORI=2042TO2047:POKEI,213:NEXT:POKE49709,215:POKE49713,213:RETURN
9520 FORI=2042TO2047:POKEI,218:NEXT:POKE49709,221:POKE49713,218:RETURN
9530 FORI=2042TO2047:POKEI,222:NEXT:POKE49709,225:POKE49713,222:LI=LI+1:RETURN
9540 FORI=2042TO2047:POKEI,225:NEXT:POKE49709,227:POKE49713,225:RETURN
9550 FORI=2042TO2047:POKEI,230:NEXT:POKE49709,233:POKE49713,230:RETURN
9560 FORI=2042TO2047:POKEI,233:NEXT:POKE49709,236:POKE49713,233:LI=LI+1:RETURN
9570 FORI=2042TO2047:POKEI,236:NEXT:POKE49709,237:POKE49713,236:RETURN
9580 FORI=2042TO2047:POKEI,238:NEXT:POKE49709,240:POKE49713,238:RETURN
9590 FORI=2042TO2047:POKEI,241:NEXT:POKE49709,243:POKE49713,241:LI=LI+1:RETURN
9600 FORI=2042TO2047:POKEI,244:NEXT:POKE49709,246:POKE49713,244:RETURN
9610 FORI=2042TO2047:POKEI,247:NEXT:POKE49709,249:POKE49713,247:RETURN
9620 FORI=2042TO2047:POKEI,250:NEXT:POKE49709,252:POKE49713,250:LI=LI+1:RETURN
9630 FORI=2042TO2047:POKEI,253:NEXT:POKE49709,255:POKE49713,253:DD=1:RETURN
9799 RETURN
9800 FORT=40TO10STEP-1:POKES1,32:POKES2,12:POKES3,T:POKES1,33
9805 PRINT"████████ WAVE";WA;"DESTROYED!!..READY FOR WAVE";WA+1
9810 PRINT"□███████                                        ":NEXT:EL=0
9898 POKEV+30,0: RETURN
9950 SYS49800:RETURN
9960 SYS49837:RETURN
9999 REM****WAVE 1****
10000 POKE53280,11:POKE53281,14
10010 PRINT"▟███████████                                  ▚         ";
10020 PRINT"                               ●●|          ";
10030 PRINT"          ●|                  ●●●●|          ";
10040 PRINT"     ●|  ●●●|      ▚         ●●●●●●|          ";
10050 PRINT"    ●●●|●●●●●|  ●●|    ●|●●●●●●●●|  ●|    ";
10060 PRINT"●|●●●●●●●●●●●|●●●●|●|●●●●●●●●●●●●|●●●|●|";
10070 FORT=0TO1:PRINT"█";A$;:NEXT
10100 FORT=0TO1:PRINT"█";A$;:NEXT
10120 FORT=0TO1:PRINT"█";B$;:NEXT:PRINT"█";A$;
10130 PRINT"█";C$:PRINT"█"
10140 PRINT"██████        ─||||||| |                                 ";
10150 PRINT"███           ─|| |                ─||| |      ";
10160 FORI=1984TO2023:POKEI,81:POKEI+54272,5:NEXT
10165 GOSUB40000
10199 RETURN
20000 RR=15:FORT=0TO15:FORTT=215TO216:POKE2040,TT:POKEV+1,PEEK(V+1)+1:GOSUB8300
20001 NEXTTT,T
20002 LI=LI-1
20005 GG=255:FORI=5TO15STEP2:GG=GG-31:POKEV+I,GG:POKEV+I-1,0:NEXT
20010 POKE2040,210:POKEV,40:POKEV+1,210:GOSUB10000:POKEV+30,0:RETURN
27000 POKEV+21,0:POKE53280,0:POKE53281,0
27005 POKES1,128:POKES1-7,32:POKES2,15:POKES2-7,15:FORT=100TO25STEP-5
27007 PRINT"▟█████████████████                    ███▙▛██   ██▛██▜"
27008 PRINT"██████████████████
27010 POKES3,T:POKES3-7,T:POKES1,129:POKES1-7,33:NEXTT:RETURN
40000 PRINT"██████████████████████████████":PRINTTAB(2)"SC";TAB(14)"LI"TAB(28)"HI";
40010 PRINTTAB(31)"█EN █ ":RETURN
40100 REM
40102 PRINT"███████████████████████████████":PRINTTAB(4)SC;TAB(16)LI;TAB(23)HI;
40105 PRINTTAB(34)EL
40110 IFSC>HITHENHI=SC
40120 RETURN
60000 PRINT"▟█████████████████████████████";:POKE53272,29:POKE53280,9:POKE53281,0
60002 POKEV+21,0:LD=4
60005 PRINT"█──────────────────────────────────────────";
60010 PRINT"□□□□     ╯/     ╯/ ╯/  ╯/ ╯/  ╯╯╯╯/  ╯/ ╯/      ";
60015 PRINT"□□     ╯●     ╯● ∨●  ╯● ╯●  ╯●●╯●  ╯●╯╯●     ";
60020 PRINT"□□     ╯●     ╯●╯╯●  ╯● ╯●  ╯● \●  ╯╯╯●●     ";
60025 PRINT"□□     ╯●     ╯╯╯●●  ╯╯╯╯●  ╯●      ╯╯●●     ";
60030 PRINT"□□     ╯●     ╯●●╯/  ╯●●╯●  ╯●      ╯╯╯●/     ";
60035 PRINT"□□     ╯●     ╯● ╯●  ╯● ╯●  ╯● ╯/  ╯●╯╯●     ";
60050 PRINT"□□  ╯╯╯╯╯/  ╯╯╯●●  ╯╯╯╯●  ╯╯╯╯●  ╯●\╯●     ";
60055 PRINT"□□  \●●●●●  \●●●   \●●●●  \●●●●  \● \●    █";
60060 PRINT"□□□□   ╯╯╯╯/ ╯/ ╯/  ╯╯╯╯/  ╯╯╯╯/  ╯/ ╯/      ";
60065 PRINT"□□   ╯╯╯╯● ╯● ╯●  \●●╯●  ╯●●●●  ╯● ╯●     ";
60070 PRINT"□□   ╯●●●● ╯● ╯●     ╯●  ╯●     ╯●╯╯●     ";
60075 PRINT"□□   ╯●     ╯╯╯╯●  ╯╯╯╯●  ╯╯/    ╯╯╯●●     ";
60080 PRINT"□□   ╯●     ╯●●╯●  ╯●●●●  ╯●●    ╯●●╯/     ";
60085 PRINT"□□   ╯●     ╯● ╯●  ╯●     ╯●     ╯● ╯●     ";
60090 PRINT"□□   ╯●     ╯╯╯╯●  ╯╯╯╯/  ╯╯╯╯/  ╯╯╯●●     ";
60095 PRINT"□□   \●     \●●●●  \●●●●  \●●●●  \●●●●     ";:GOSUB8300:EN=10
60096 PRINT"████████████████████████████████      MUSIC █ON ---BY F.G.TOUT 1984":LD=2
60097 SYS36864:PRINT"██--SPACE TO PLAY----K FOR KEY REFERENCE--"
60100 PRINT"███ █   LEVEL █";LD;"█  ENEMYS PER WAVE ";EN;"█  "
60105 GETZ$:IFZ$=""THEN60105
60110 IFZ$="█"THENPOKES1,128:POKES2,10:POKES3,150:POKES1,129:LD=LD+1
60112 IFZ$="█"THENPOKES1,32:POKES2,10:POKES3,150:POKES1,33:EN=EN+1
60116 IFZ$="K"THENPOKES1,128:POKES2,10:POKES3,15:POKES1,129:GOTO62000
60117 IFZ$="█"THENPOKES1,32:POKES2,12:POKES3,6:POKES1,33:LD=2:EN=10
60118 IFZ$="█"THENSYS36864:PRINT"████████████████████████████████      MUSIC █ON "
60119 IFZ$="█"THENSYS36864+256:PRINT"████████████████████████████████      MUSIC █OFF"
60120 IFZ$=" "THENGOSUB63000:POKE49607,LD:RETURN
60125 IFLD>17THENLD=2:PRINT"███████████        "
60130 GOTO60100
62000 POKE53281,2:POKE53280,10:PRINT"▟";:POKE53272,22
62010 PRINT"████[—.1] INCREASE ♠PEED OF ♣LIENS"
62020 PRINT"███▙[—.3] ╮NCREASE ♣LIEN ╱UMBER"
62030 PRINT"████[—.5] \USIC ┌N"
62040 PRINT"███▙[—.7] \USIC ┌FF"
62050 PRINT"████[—LR/ |OME]  ▁ESET"
62060 PRINT"███▙[♠PACE] |O ▁ETURN"
62070 GETW$:IFW$=""THEN62070
62075 IFW$=" "THEN60000
62080 GOTO62070
63000 FORT=54272TO54296:POKET,0:NEXT:POKE54296,15
63005 POKES2,15:POKES2-7,15:POKES1,32:POKES1-7,32:POKES1-14,0
63010 READV1,V1,DU:FORT=0TODU+150:NEXT
63020 IFV1=0THENRESTORE:RETURN
63030 POKES3,V1:POKES3-7,V2:POKES1,33:POKES1-7,33
63040 GOTO63010
63050 DATA5,5,50,10,10,60,15,15,40,10,10,30,5,5,50,4,4,50,3,3,50,5,5,50,0,0,0
READY.
```



Awesome *in its conception*

Brilliant *in its depiction*

Dynamic *in its execution*

The world you are about to enter bears no resemblance to any arena you ever encountered before. Weapons are of no avail in this small habitat.

The only sources of protection at your disposal are quick wits and fast reflexes. The only reward is to survive against monsters of unbelievable ferocity and cunning, and to avoid hazards more perilous and deadly than any you might find on a trip through the outer universe. This is the real world, populated by the creatures of our own inner universe, where nature is red in tooth and claw. Brace yourself now, and come with Starcade into the still water and deceptive calm of the **SAVAGE POND.**

COMMODORE, ATARI, BBC/ELECTRON (£8.95 each)
On disc for COMMODORE, ATARI, BBC/ELECTRON (£10.95 each)

Now available for SPECTRUM (£7.95) ***From Boots and all leading retailers***



Up Up and Away

COMMODORE, ATARI, BBC
(£8.95 each)
On disc for COMMODORE, ATARI,
BBC (£10.95 each)

STARCADE

Argus Press
Software Group

produced by
ARGUS PRESS SOFTWARE
No 1 Golden Square, London W1R 3AB
Telephone: 01 437 0626

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written an exhilirating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications Limited
No 1 Golden Square
London WIR 3AB

★PLEASE COMPLETE IN BLOCK CAPITALS

Your Name

Program Name

Computer/memory size it runs on

Amount of memory program occupies

Other computers/memory size which your program runs on without conversion or use

Does your game need or use joysticks? Yes No

Have you sent your game to another magazine Yes No

Is it original/or a variation on a theme?

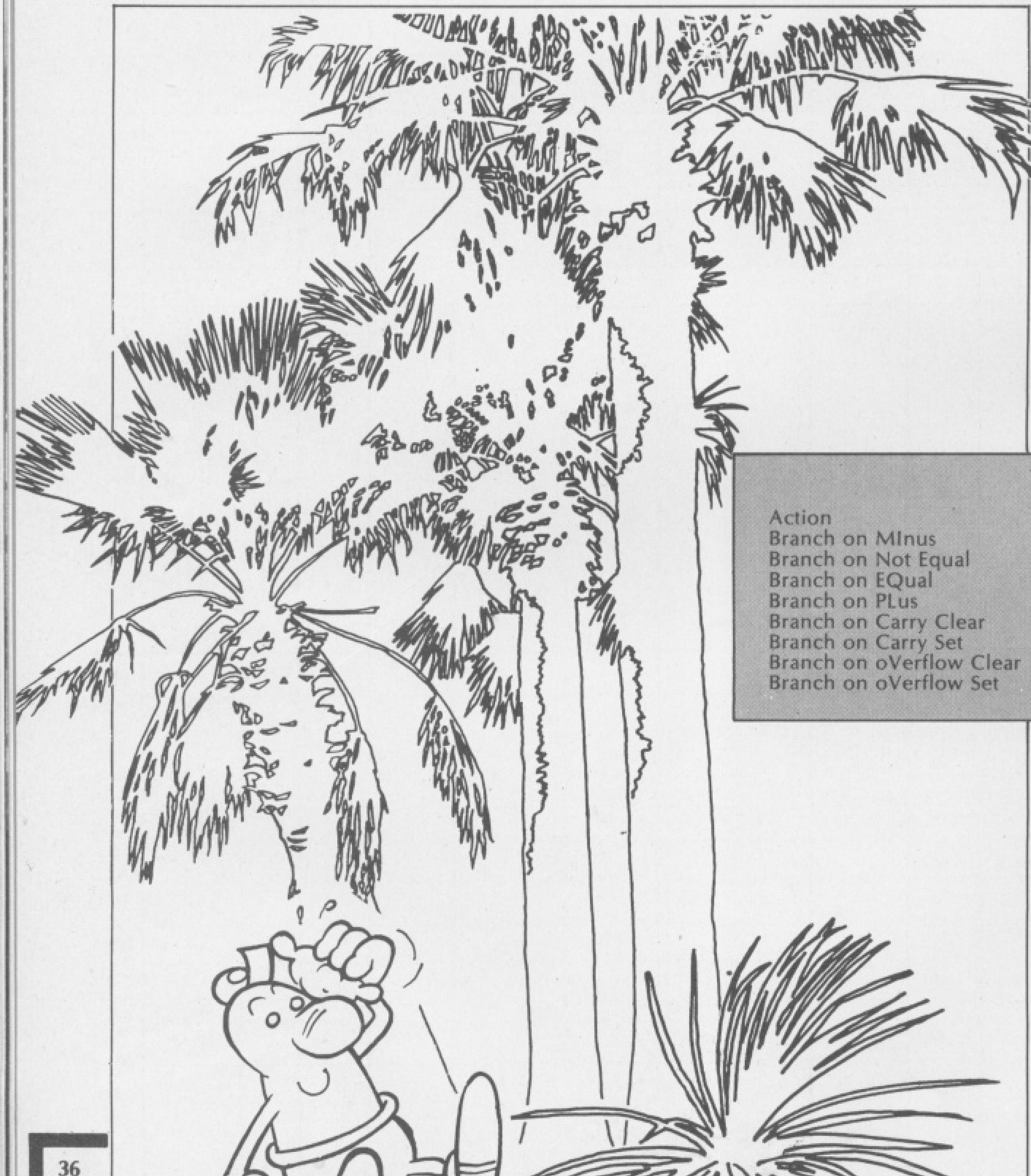
Your Address

Telephone Number

Times to contact you

Branching and the status register are the subjects of this month's installment of our machine code series from A.P. and D.J. Stephenson.

MASTERING MACHINE CODE



WHEN PROGRAMMING IN BASIC, the IF/THEN statement provides an easy way of introducing a **decision.** The conditions can be quite complex, such as:

```
100 IF A < 3★(B+D) OR C=0
THEN PRINT "WRONG":
GOTO 350
```

Alas, it is not so easy in machine code. In the first place, the decision options are known as **branch** instructions and there are only eight to choose from as can be seen from Table 4.1. As usual, xx is used to indicate the operand is **one byte** long and can be specified by two hex digits.

TABLE 4.1

Action	Assembler	Hex code
Branch on MInus	BMI xx	30 xx
Branch on Not Equal	BNE xx	D0 xx
Branch on EQual	BEQ xx	F0 xx
Branch on PLus	BPL xx	10 xx
Branch on Carry Clear	BCC xx	90 xx
Branch on Carry Set	BCS xx	B0 xx
Branch on oVerflow Clear	BVC xx	50 xx
Branch on oVerflow Set	BVS xx	70 xx

What does it mean when, for example, we write BNE? We can see from the table above that it means 'branch if not equal' but it immediately poses the question — branch if WHAT is not equal? The answer to this is — if the Z bit in the process status register is 0. It would help you at this point if you turn back to Part 2 of this series where you will find details of the flags in this register and the conditions under which they are set to 1 or cleared to 0. You will see that an appropriate flag is updated automatically after most of the instructions. So, still referring to BNE as an example, the branch will take place only if the last instruction caused a non-zero result. That is to say, the microprocessor will examine its process register to see if the Z bit was a 0. The opposite instruction, BEQ can



be used to test if the last result was zero — the branch will now take place if the Z bit is 1.

Another paid of branch instructions is BPL and BMI. A branch takes place with BMI only if the N bit is 1. You will remember from Part 1 of the series that single byte negative numbers in two's complement notation always have a 1 in the msb position but positive numbers always have a zero. So, if you are working in two's complement (but you won't always be), you can use either BPL or BMI as appropriate, to test the sign of the last result. It is worth mentioning here that these two instructions may not be used as frequently as the other branch instructions.

BCS and BCC are another pair of opposites. These are used to test if the last operation caused a result which was 'too big' to be handled by a single byte, so forcing a 'carry out'. This **carry bit** is captured in the C bit position of the status register. With BCS, a branch takes place only if the C bit is 1. With BCC, a branch takes place only if the C bit is 0.

The final pair of opposites are BVC and BVS. A branch takes place with BVS only if the V bit is 1. In other words, you can test if the last result caused **two's complement overflow** because, if it did, the V bit is set to 1. However, 'two's complement overflow' must not be confused with the carry-out condition mentioned earlier. Indeed, it is possible for the overflow condition to exist without a carry out occurring and vice versa. In case this seems strange to you, consider what happens if we add 1 to +127 in two's complement binary:

```
+127        0111 1111
add 1               1
-----------------------
result      1000 0000
```

If we treat the result (128 in decimal) as a pure binary absolute number, it is quite sensible. If, on the other hand, we interpret the result in two's complement, it is clearly absurd because 1000 0000 is -128. Although there has been no carry out, an overflow condition is established which would set the V bit to 1. The largest positive two's complement number which can be held in a single byte is 127 so, if we try to add 1, we must expect the result to be invalid — overflow in fact. This is a tricky business so it is worth one more example:

```
   -2       1111 1111
add +1      0000 0001
   --------------------
   1        0000 0000 result
```

If we add plus 1 to -1, the result should be zero, as indeed the above shows. There is therefore no overflow condition and yet there is a carry out — which is ignored! So, the V bit would show a 0 but the C bit would show a 1 after this example. This should illustrate the point we are trying to make; it is the way we **interpret** the arithmetic that determines whether we ignore the C bit or the V bit. It must be firmly established in your mind that 'overflow' is a condition which only makes sense if you are conducting arithmetic in two's complement form. If you are working with **absolute** numbers only (all positive) the concept of overflow, as signalled by the V bit, has no meaning. Later in the series, when we deal with peripheral controls, we shall learn that the V bit is employed in another role, quite unrelated to overflow.

Calculating branch operands

We have, in the preceeding paragraph, frequently men - tioned that 'the branch takes place if — etc etc'. The next topic of discussion is how we work out the branch destination. In BASIC, of course, we just say GOTO followed by a line number. We can't do this in machine code because the concept of line numbers does not exist. (Even if we had an assembler fitted, we still couldn't use a line number as a branch destination.) The operand in a branch instruction is a number which, if the condition is true, informs the microprocessor how many bytes forward (or backward) the next instruction is to be found. This is called **relative addressing** because it indicates a destination address relative to the present address. Only the eight branch instructions use relative addressing.

To understand this mode of addressing, we should be clear in our minds as to the role of the most important register in the microprocessor, the **Program Counter** (PC). This is a 16-bit register which always contains the absolute address of the NEXT **instruction byte** to be executed. As you already know, a machine code program is simply a set of bytes, stored in memory in sequential addresses. Suppose these program bytes are stored in a block from address $C000 (49152 decimal) onwards. To execute the program, we can simply enter SYS 49152 and press RETURN. This will cause the hex equivalent of 49152 to be loaded into the program counter. The sequence of events is then entirely automatic: the instruction byte at address 49152 is brought into the microprocessor and processed, after which the program counter goes up 1 and the next sequential byte is processed and so on. However, if a branch instruction is encountered and conditions are true, the operand byte of the instruction is added to the present contents of the program counter. The smooth sequential action is now replaced by a sudden jump to a new instruction byte address. Once the new address has been reached, the program counter proceeds in orderly sequential fashion from that new address. An example will help:

1. Assume the program contains a set of bytes located from address $C000 onwards.
2. Now assume the program has reached the second byte of the instruction BNE $07 located at address $C011. The program counter will then contain $C012 — the next instruction byte address.
3. If the branch conditions



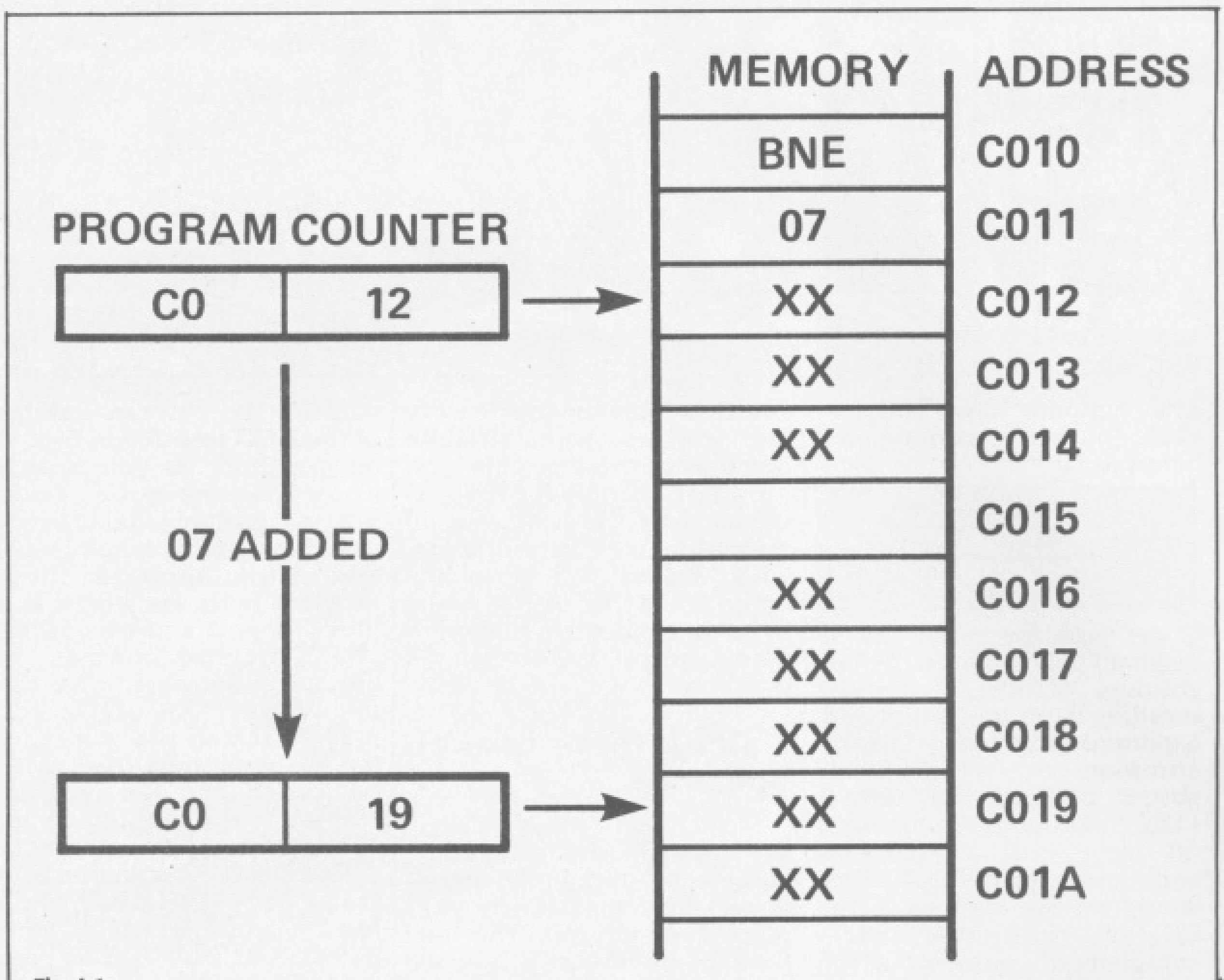


Fig 4.1

are true and the branch is taken, the operand ($07) will then be added to the program counter which will then contain $C019. The next instruction byte to be executed will then be taken from this address — seven bytes forward. If the branch conditions were not true, the program counter would not be altered and the next instruction byte at $C012 would be executed.

Figure 4.1 illustrates our example and should be carefully studied.

Calculating backward branches

As you are aware when programming in BASIC, GOTOs can be to higher or lower line numbers. How do we calculate the operand number if we wish to branch backwards? The answer is that we resort to two's complement arithmetic. Relative addressing can be used with negative numbers in the operand to indicate a branch back to an earlier address. For example, if we wish to branch 7 bytes backwards, we must have an operand of -7. This is where our previous knowledge of two's complement arithmetic is brought into use. If you understood Part 1 of the series, you will be able to work out that -7 can be calculated as follows:

```
+7    0000 0111
-7    1111 1001 = $F9
```

Thus if we write, say, BNE $F9, and the condition is true, it will cause 7 to be subtracted from the program counter. Figure 4.2 illustrates this example.

It should be realised, from previous discussions on two's complement, that the maximum number of forward bytes which can be used with any branch is 127 and the maximum backward bytes, 128. This is because the operand of a branch instruction can only be **one byte** long.

Fortunately, it is very unusual in practical programming to require branches greater than these allowed limits. However, as we shall see later, there is a way of overcoming the problem, should it ever arise.

Mistakes in machine code programs can often be traced to incorrect branch operands because it is so easy to be one out in the byte count. Another pitfall is the status register flags. When a branch instruction is encountered, the current state of the flags determine whether or not the branch takes place. Normally, but not always, it is the effect of the **last** instruction which is being tested. However, some instructions do not effect the status flags, in which case, the test is dependent on earlier conditions.

The following instructions have no effect on the status flags: STA, STX, PHA, PHP, TXS and all the branch instructions. The fact that branch instructions have no effect on status flags means that branch instructions can follow one another in order to test for two different conditions.

The status flags effected depend on the instruction so it is important to be familiar with Table 4.2, showing which of the status flags are affected.

At this stage, of course,

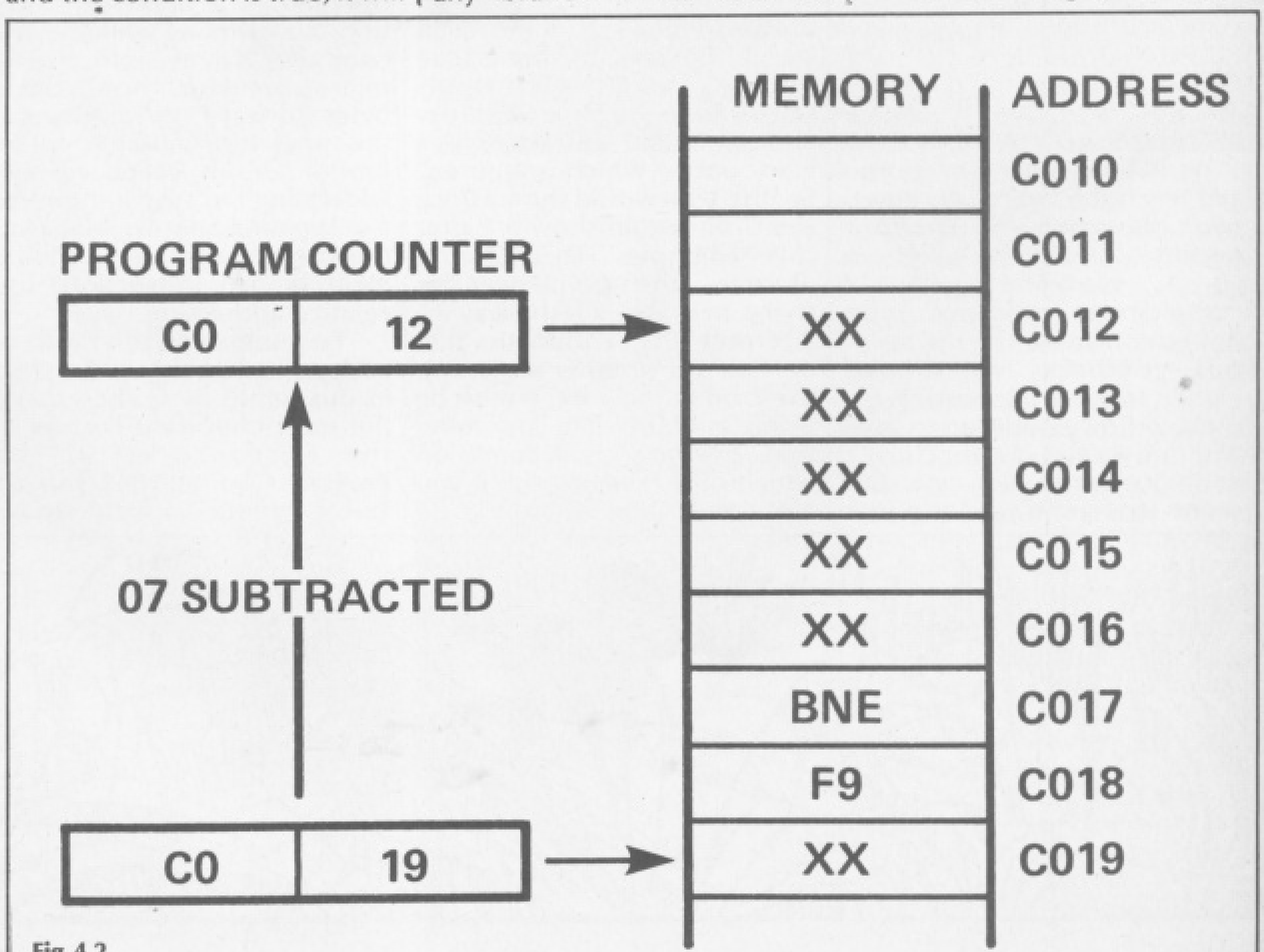


Fig 4.2

Table 4.2 Flags effected

Updates N,Z and C flags: ADC,ASL,CMP,CPX,CPY,ROL, ROR,SBC

Updates N and Z flags: AND,DEC,DEX,DEY,EOR,INC,INX, INY,LDA,LDX,LDY,ORA,PLA,TAX,TAY,TYA,TSX,TXA

Updates N,Z,C and V flags: ADC,SBC.

Updates N,C and always clears N to 0 LSR

many of the above instructions have not yet been discussed but the table will be helpful to refer back to. It shows, for example, that there is little point in using BCC after DEX to check if it had reset the carry to zero because only the N and Z flags are effected by DEX. If the carry was 0, it would have been due to the effects of an earlier instruction.

Branching when an assembler is fitted

Calculating the number of bytes in the operand of a branch instruction is fraught with danger. It is so easy to get it wrong, particularly with backward branches. If the count is out by one, it is highly probable the program will crash because the branch destination could be the **operand** instead of the **op-code** of an instruction. If by chance, the operand happens to be a legitimate op-code, the microprocessor will treat it as such and execute it! If, on the other hand, it doesn't recognise it, the program would crash.

One of the most useful aspects of an assembler is the way it allows us to write branch instructions. Instead of counting the bytes, we can use a label of our choice. Using the MIKRO assembler, but using xx to represent example bytes, a branch would look something like:

```
        BNE  VOLTS
        xx   xx
        xx   xx   xx
        xx   xx
VOLTS   xx   xx
        xx   xx   xx
        etc  etc
```

The operand lable in the branch has been chosen as VOLTS. The assembler will search down the program until it finds a line beginning with VOLTS. Note that the assembler demands that the destination label ends with a space. As an excercise, satisfy yourself that, without an assembler, the above branch instruction in hex machine code would be, D0 07. The op-code for BNE is D0 and the branch is 7 bytes forward — not 8.

As a simple illustration, study the following few lines written in assembly language and, for comparison purposes, in hex machine code.

Assembly	Hex coding
LDA $06	A9 06
BNE JOE	D0 04
STA $C400	8D 00 C4
BRK	00
JOE STA $C401	8D 01 C4
BRK	00

It is a trivial 'program' and it is not expected that you try and run it. It starts by loading 06 into the accumulator, using immediate addressing. This will make the Z bit=0. It then branches to the line labelled JOE. It must do, of course, since the accumulator is certainly not zero. The accumulator contents are then stored in the absolute address $C401. The computer then reaches BRK (which means BReak and is roughly equivalent to END in BASIC). If the first line of the program was changed so that the accumulator was initially loaded with 00, the branch would not be taken and the accumulator contents would then be stored in $C400 before reaching BRK. So, depending on the state of the accumulator, the program will either stop in the middle or at the end. Examine the hex code version carefully, particularly the branch operand — satisfy yourself that this is correct. Note also that the op-code for BRK is 00. You will no doubt agree that the assembler version is easier and more informative.

Programming the status register

When we discussed the status register bits in connection with branch instructions, it was established that the relevant bits are **automatically** updated by the microprocessor after each instruction. However, there are certain situations where it is necessary for the programmer to intervene and manually alter the bits. The seven instructions available for this purpose are shown in Table 4.3

Table 4.3 Status register instructions

Action	Assembler	hex code
Clear Carry	CLC	18
SEt Carry	SEC	38
CLear Decimal	CLD	D8
SEd Decimal	SED	F8
CLear oVerflow	CLV	B8
CLear Interrupt mask	CLI	58
SEt Interrupt mask	SEI	78

Note we can set or clear the C bit, or clear the V bit. We shall not attempt, at this stage, to discuss the conditions under which these instructions should be used. Neither would it be profitable to discuss interrupt masks or D bits. For the moment, treat Table 4.3 as reference material to be consulted later.

Comparison instructions

There are times when a simple branch test is not quite what we want. For example, BEQ and BNE can only test for zero or non-zero. What if we want to find out if a register contains some particular number, 7 perhaps? One way which comes to mind is to subtract 7 from the number in the register and then use BNE or BEQ to see if the result is zero. Obviously, if X–7=0 then X must contain 7. Unfortunately, the subtraction operation destroys the number originally in the register and we may not want this — it would be destructive testing. We could, of course, save the register contents somewhere before the test and re-load again afterwards but this, to say the least, would be cumbersome. Fortunately, there are three beautiful comparison instructions available which perform the subtraction for you but without hurting the register you are testing. It is called 'transparent' subtraction. We shall describe the action of CMP first and, because it is the easiest to understand, use the immediate addressing mode for example purposes.

Suppose we write CMP $07. This will compare the hex number 07 with the number in the accumulator (A) and update the Z,N and C bits according to the following rules:

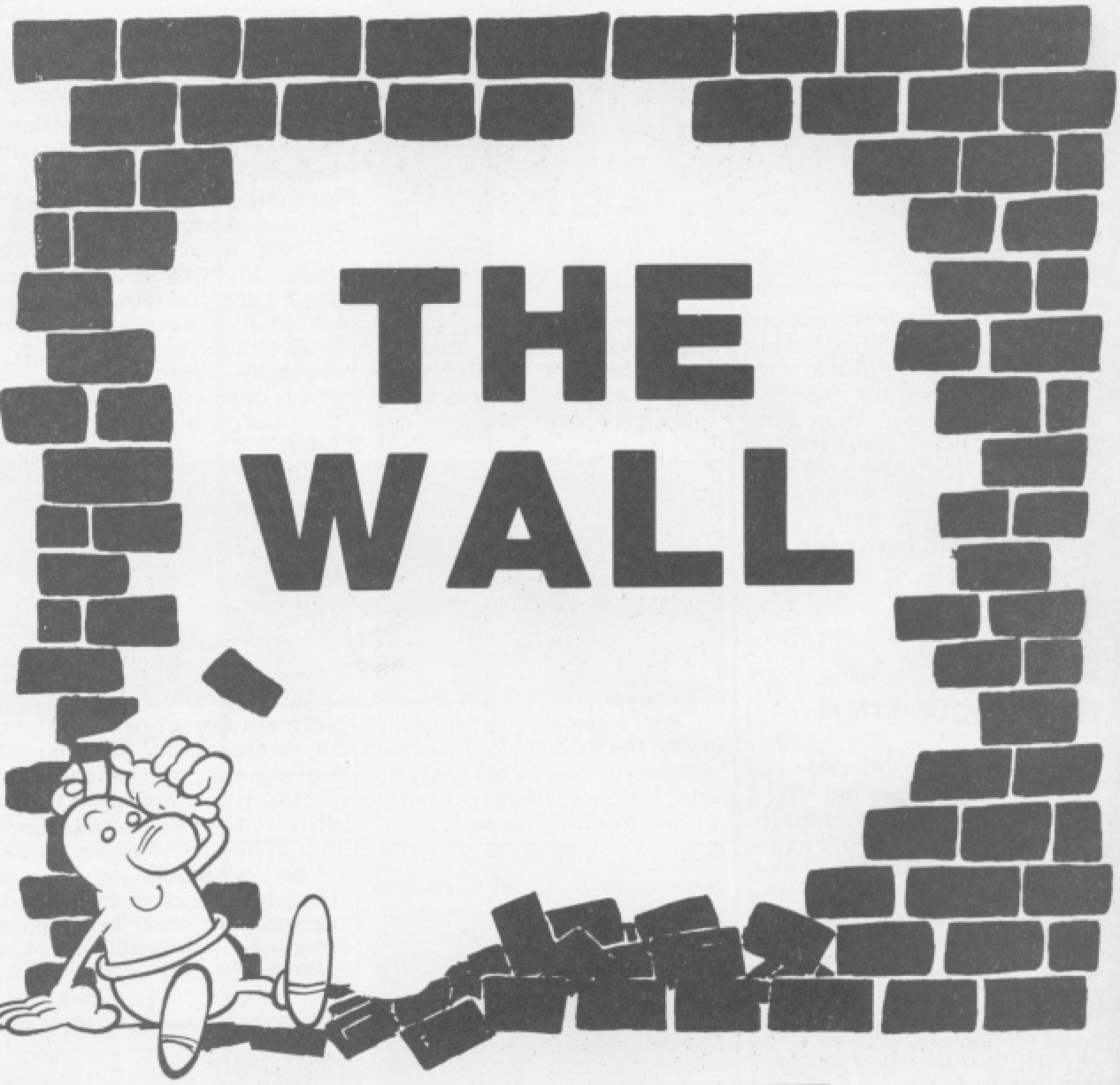
1. If operand number ($07 in this case) = number in register, Z and C become 1
2. If operand number is less than number in register, C becomes 1 and Z becomes 0
3. If operand number is greater than number in register, C and Z become 0
4. If operand number is less than or equal to number in register, C becomes 1. It should be clearly understood that a compare instruction only effects the status flags. It does nothing else. Therefore, the only possible instruction after CMP is a suitable branch test. Unless you use a branch immediately afterwards, there is no point in using CMP in the first place.

Although we have used immediate addressing in our example, a wide range of addressing modes are possible. Thus, we can compare the contents of a particular memory address. The full range of comparison instructions are shown in Table 4.4.

Table 4.4 The comparison instructions

	Assembler	Hex code
Compare A	CMP # $xx	C9 xx
	CMP $xx	C5 xx
	CMP $xxxx	CD xx xx
	CMP $xx,X	D5 xx
	CMP $xxxx,X	DD xx xx
	CMP $xxxx,Y	D9 xx xx
	CMP ($xx,X)	C1 xx
	CMP ($xx),Y	D1 xx
Compare X	CPX # $xx	E0 xx
	CPX $xx	E4 xx
	CPX $xxxx	EC xx xx
Compare Y	CPY # $xx	C0 xx
	CPY $xx	C4 xx
	CPY $xxxx	CC xx xx

Try and stop poor Pinky from being driven completely up the wall in this game from F.G. Tout.



'THE WALL' RUNS IN BASIC with several machine code routines to move all 8 sprites, change sprites and rotate the colours at the End Of Game sequence.

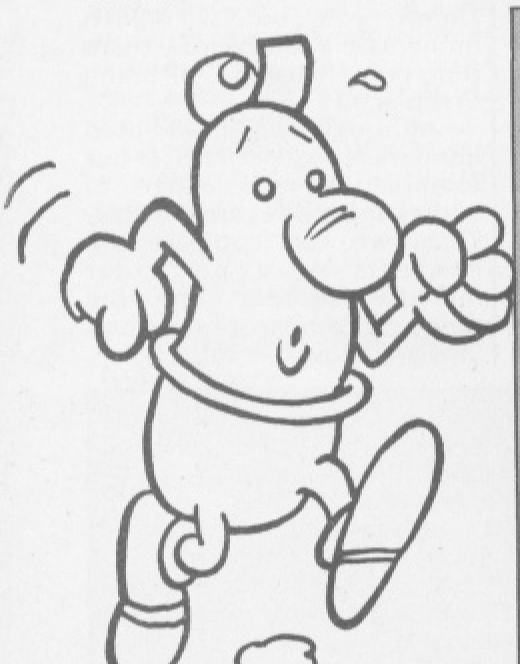
For those who know about Pinky, this will be quite familiar, for those who don't, Pinky is a young man sinking slowly into a state of total insanity!...and to top things, everybody's after him — the wife, the school-teacher, the mother, the lawyer and the hammers.

Can you save Pinky by guiding him through 5 waves of walls and ladders?

The screen is a wrap-around screen so you can jump from one side and appear on the other, but you can still collide off screen, so be careful.

When you get Pinky to the canes at the top of the screen you will proceed to the next wave.

Type in parts 1, 2 and 3 separately, and save before running. Then, when all is well, plug your joystick into port 2 and you can start to play...THE WALL.



Program Information

Part 1

This is quite straight forward: it's all sprite data.

Part 2

1 — 100	Download characters
6014 — 50030	Machine code move me...
60014 — 63025 —	Machine code move and change enemy sprites
63030 — 63032	Machine code rotate colours
63400 — 63850	Instructions...

Part 3

1 —	Sound variables/clear sound
2 — 520	Variables and sprite positions
2000 —- 2010	Time limit
3000 —- 3000	Main routine
4000 —- 5020	Check collision with wall
6010 —- 9605	Move me and jump
10000 — 10010	Alter enemy sprite positions.
13000 — 13060	Hit enemy?
14000 — 14025	Game over
15000 — 15099	Screen 1
16000 — 16099	Screen 2
17000 — 17099	Screen 3
18000 — 18099	Screen 4
19000 — 19099	Screen 5
60000 — 60099	Update score
63000 —	Title page

Variables

A0 - A9	Machine code address
S1, S2, S3	Sound
V	Sprite variable
LI	Lives
LE	Level
SC	Score....HI high score
B1	Decrement time limit
C1-C5	Sprite pos.
Q	Change me



Program Listing Part 1

```
565 FORI=12928TO15615-192:READA:POKEI,A:NEXT
6000 POKE198,2:POKE613,13:LOAD
60000 DATA63,255,252,63,255,252,63,255,252,63,255,252,63,255,252
60010 DATA48,126,12,0,126,0,0,126,0,0,126,0,0,126,0,0,126,0,0,126,0,0,126,0
60020 DATA0,126,0,0,126,0,0,126,0,0,126,0,0,126,0,0,126,0,0,126,0,0,255,0,0
60030 DATA127,129,254,63,0,252,63,0,252,63,0,252,63,0,252,63,0,252
60040 DATA63,0,252,63,129,252,63,255,252,63,255,252,63,255,252,63,255,252
60050 DATA63,129,252,63,0,252,63,0,252,63,0,252,63,0,252,63,0,252,63,0,252
60060 DATA63,0,252,127,129,254,0
60062 DATA127,255,252,63,255,252,63,255,252,63,255,252,63,0,12,63,0,4,63,0,0
60070 DATA63,128,0,63,252,0,63,252,0,63,252,0,63,252,0,63,128,0,63,0,0
60080 DATA63,0,4,63,0,12,63,255,252,63,255,252,63,255,252,63,255,252
60085 DATA127,255,252,0
60090 DATA127,1,252,62,0,248,62,0,248,62,0,248,62,0,248,62,0,248,62,0,248
60095 DATA62,0,248,62,124,248,62,124,248,62,124,248,62,124,248
60100 DATA62,124,248,62,124,248,62,124,248,63,255,248,63,255,248,63,255,248
60110 DATA63,255,248,63,239,248,127,199,252,0
60120 DATA127,255,252,63,255,248,63,255,248,63,255,248,63,131,248,63,1,248
60130 DATA62,0,248,62,0,248,62,0,248,63,255,248,63,255,248,63,255,248
60140 DATA63,255,248,62,0,248,62,0,248,62,0,248,62,0,248,62,0,248,62,0,248
60150 DATA62,0,248,127,1,252,0
60160 DATA127,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0
60170 DATA62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,63,255,248
60180 DATA63,255,248,63,255,248,63,255,248,127,255,248,0
60260 DATA127,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0
60270 DATA62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,62,0,0,63,255,248
60280 DATA63,255,248,63,255,248,63,255,248,127,255,248,0
60290 DATA0,0,0,7,255,255,9,36,147,31,255,253,36,146,77,73,36,149,255,255,229
60292 DATA146,73,37,146,73,37,146,73,39,159,201,37,159,201,45,146,73,53
60294 DATA146,73,37,146,73,37,159,201,39,146,73,38,146,73,44,146,73,56
60296 DATA255,255,240,255,255,224,0
60298 DATA0,44,0,0,187,0,0,238,0,0,59,0,0,90,0,0,58,0,0,20,0,0,60,0,0,58,0,0
60300 DATA174,128,6,184,128,6,44,64,0,56,60,0,40,0,0,43,192,0,46,192,0,32,192
60302 DATA0,48,192,0,48,0,0,48,0,0,240,0,0
60304 DATA0,44,0,0,187,0,0,110,0,0,59,0,0,94,0,0,11,0,0,20,0,0,60,0,0,40,0
60306 DATA0,174,0,0,186,0,0,174,0,0,125,0,0,105,0,0,40,0,0,40,0,0,40,0,0,60,0
60308 DATA0,60,0,0,60,0,0,252,0,0
60310 DATA0,44,0,0,187,0,0,110,0,0,27,0,0,94,0,0,27,0,0,20,0,0,60,0,0,44,0
60312 DATA0,44,0,0,60,0,0,120,0,0,104,0,0,40,0,0,168,0,2,170,0,3,10,0,3
60314 DATA3,0,15,3,0,0,3,0,0,15,0,0
60316 DATA0,44,0,0,187,0,0,110,0,0,59,0,0,94,0,0,27,0,0,20,0,0,60,0,0,174,0,2
60318 DATA174,128,6,60,144,4,56,16,0,40,0,0,40,0,0,168,0,3,139,0,15,11,192
60320 DATA60,3,192,0,0,192,0,0,192,0,0,192,0
60322 DATA0,56,0,0,238,0,0,187,0,0,228,0,0,181,0,0,228,0,0,20,0,0,60,0,0,174
60324 DATA0,2,186,128,2,44,144,1,56,16,1,44,0,0,56,0,2,232,0,3,232,0,3
60326 DATA8,0,3,12,0,0,12,0,0,12,0,0,15,0,0
60328 DATA0,56,0,0,238,0,0,187,0,0,228,0,0,181,0,0,224,0,0,20,0,0,60,0,0,172,0,0
60330 DATA186,0,0,174,0,0,186,0,0,109,0,0,121,0,0,40,0,0,40,0,0,40,0,0,60
60332 DATA0,0,60,0,0,60,0,0,63,0,0
60334 DATA0,56,0,0,238,0,0,187,0,0,228,0,0,181,0,0,228,0,0,20,0,0,60,0,0,44,0,0
60336 DATA168,0,1,186,64,1,62,64,0,40,0,0,56,0,0,42,0,0,34,0,0,34,192,0,160,192
60338 DATA0,192,240,0,192,0,0,240,0,0
60340 DATA0,56,0,0,238,0,0,187,0,0,228,0,0,181,0,0,36,0,0,20,0,0,56,0,0,174,0,2
60342 DATA186,128,6,44,144,4,56,16,0,40,0,0,56,0,0,42,0,0,162,192,3
60344 DATA192,176,15,0,48,12,0,60,12,0,0,0,0,0,0
60346 DATA0,56,0,0,238,0,0,187,64,0,238,64,0,187,128,1,44,128,1,20,128,2,190
60348 DATA128,2,170,0,0,60,0,0,40,0,0,60,0,0,60,0,0,40,0,0,40,0,0,44,0,0
60350 DATA44,0,0,60,0,0,48,0,0,48,0,0,48,0,0
60352 DATA0,56,0,0,238,0,0,187,0,0,238,0,0,187,0,1,44,64,1,20,64,2,190,128,2
60354 DATA170,128,0,60,0,0,40,0,0,56,0,0,44,0,0,56,0,0,40,0,0,40,0,0,40,0,0
60356 DATA60,0,0,60,0,0,60,0,0,60,0,0
60358 DATA0,56,0,0,238,0,1,187,0,1,238,0,2,187,0,2,44,64,2,20,64,2,190,128,0,170
60360 DATA128,0,60,0,0,40,0,0,60,0,0,40,0,0,60,0,0,40,0,0,56,0,0,56,0,0,60,0,0
60362 DATA12,0,0,12,0,0,12,0,0
60364 DATA0,8,0,12,46,12,195,184,48,52,229,67,4,180,204,203,37,51,58,208,204
60366 DATA14,172,48,3,186,192,48,235,176,15,172,156,0,243,19,3,176,192,204,171
60368 DATA48,51,56,204,12,42,3,48,226,204,195,40,192,12,12,240,0,12,3,0,15,0,0
60370 DATA0,40,0,48,174,0,12,180,0,3,229,48,48,148,12,12,16,48,3,40,204,48,175
60372 DATA48,14,186,144,6,43,144,4,236,12,0,59,0,3,56,192,204,235,48,3,56,204
60374 DATA12,43,0,48,35,48,3,51,0,12,51,204,0,48,0,0,60,0,0
60376 DATA0,34,32,0,8,128,12,193,16,61,197,16,253,69,80,207,85,16,203,149,16,207
60378 DATA245,0,75,133,0,79,196,192,74,133,208,10,133,24,10,129,10,8,161,2,8,104
60380 DATA2,8,24,8,8,40,8,8,160,32,8,128,128,8,242,0,15,8,0,0
60382 DATA0,34,32,0,8,128,12,193,16,61,197,16,61,69,80,63,85,16,59,149,16
60384 DATA63,245,0,27,133,216,7,196,218,6,133,2,10,133,10,10,129,8,8,161,32
60386 DATA8,40,128,8,10,0,40,2,128,32,0,128,32,0,240,32,0,0,60,0,0,0
60388 DATA32,34,32,128,8,128,140,193,16,173,197,16,45,69,80,47,85,16,43,149,16
60390 DATA9,245,0,13,69,0,15,196,0,10,133,0,10,133,0,10,129,0,8,129,0,8,128,0,8
60391 DATA128,0,232,128,0,224,128,0,192,128,0,0,128,0,0,240,0,0
60398 DATA0,0,0,0,0,0,0,0,3,80,0,141,52,0,39,92,0,129,180,0,0,28,0,0,116,0,1
60400 DATA208,12,7,64,54,29,0,28,52,0,112,220,0,208,116,0,112,220,1,192,116,71
60402 DATA64,29,221,192,55,119,0,13,220,0,3,48,0,0
60404 DATA0,0,48,0,0,224,112,0,232,56,0,10,54,0,1,1,128,11,0,108,228,0,27,144
60405 DATA0,15,128,0,62,64,0,242,0,43,193,0,235,2,0,20,9,128,215,6,64,60,56,128
60406 DATA150,44,176,60,32,160,195,32,40,0,32,8,0,240,60,0
60408 DATA0,0,0,0,12,0,0,56,0,0,58,12,0,1,57,128,11,58,108,230,0,31,152,0,63
60410 DATA128,0,242,64,43,194,0,235,1,0,20,2,0,215,9,128,60,6,64,150,56,128,60
60412 DATA44,160,195,32,140,0,160,10,0,128,2,3,192,15,0
60414 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,12,0,0,58,108,230,58,191,154,0,255
60416 DATA130,43,194,66,235,2,15,20,1,3,215,2,0,60,9,128,150,6,64,60,8,128,195
60418 DATA12,128,0,8,128,0,8,128,0,8,128,0,63,192,0
60420 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,64,0,24
60422 DATA128,0,32,0,0,16,0,0,32,0,0,32,1,255,47,253,170,47,253,85,16,1,255,32
60424 DATA0,0,152,0,0,0
60426 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,64,0,6,80,0,8,32,0,28,16,0,47,0
60428 DATA0,19,196,0,160,247,0,100,55,192,24,6,240,4,5,188,0,3,111,0,0,218,0,0
60430 DATA53,0,0,12,0,0,0,0
60432 DATA0,0,0,0,32,0,0,32,0,0,38,153,0,38,153,0,32,194,0,32,193,0,0,194,0,0
60434 DATA201,0,0,192,0,0,192,0,0,192,0,0,192,0,1,80,0,1,224,0,1,224,0,1,224,0
60436 DATA1,224,0,1,224,0,1,224,0,1,224,0
60438 DATA0,168,0,3,19,0,3,19,0,0,184,0,2,206,0,2,118,0,0,16,0,3,187,0,6,238,64
60440 DATA4,184,64,4,236,64,4,184,64,7,171,64,6,254,64,3,171,0,2,254,0,0,68,0
60442 DATA0,68,0,0,68,0,0,68,0,3,207,0,0
60446 DATA0,0,0,0,168,0,3,19,0,3,19,0,0,184,0,2,206,0,2,118,0,0,16,0,3,187,0,6
60448 DATA238,64,4,184,64,4,236,64,4,184,64,7,171,64,6,254,64,3,171,0,2,254,0
60450 DATA193,68,0,197,4,0,121,4,0,208,15,0,0
60456 DATA0,0,0,0,168,0,3,19,0,3,19,0,0,184,0,2,206,0,2,118,0,0,16,0,3,187,0,6
60458 DATA238,64,4,184,64,4,236,64,4,184,64,7,171,64,6,254,64,3,171,0,2,254,0
60460 DATA1,69,12,0,65,76,0,64,92,3,192,28,0
60462 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,48,0,0,60,0,0,191,0,0,175,192,0,219
60464 DATA240,0,186,170,96,221,221,136,170,170,106,63,255,0,63,252,0,63,240,0
60466 DATA63,192,0,63,0,0,60,0,0,48,0,0,0,0,0,0
60468 DATA0,0,0,0,0,0,0,0,0,48,0,0,60,0,0,63,0,0,63,192,0,191,240,0,191,252,0
60470 DATA191,255,0,170,170,160,170,170,170,170,170,160,191,255,0,191,252,0,191
60472 DATA240,0,63,192,0,63,0,0,60,0,0,48,0,0,0,0,0,0
60474 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,0,0,224,0,0,168,0,0,230
60476 DATA102,96,85,85,132,191,255,169,21,84,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
60478 DATA0,0,0,0,0,0,0
60480 DATA0,168,0,2,86,0,2,18,0,2,86,0,2,86,0,0,152,0,0,48,0,0,48,1,3,187,9,15
60482 DATA171,232,63,171,224,127,171,192,79,3,192,15,171,192,3,171,192,3,187,192
60484 DATA3,187,192,1,123,0,0,56,0,0,8,0,0,5,0,0
60486 DATA0,0,0,0,168,0,2,86,0,2,18,0,2,86,0,2,70,0,0,152,0,0,48,0,122,187,0,127
60488 DATA171,233,63,171,233,15,171,192,15,3,192,15,171,192,3,171,192,3,187,192
60490 DATA3,187,192,0,187,0,0,184,0,0,136,0,1,69,0,0

READY.
```

Program Listing Part 2

```
1 POKE56334,PEEK(56334)AND254
2 POKE1,PEEK(1)AND251
3 FORG=0TO64*8:POKE12288+G,PEEK(53248+G):NEXT
4 POKE1,PEEK(1)OR4
5 POKE56334,PEEK(56334)OR1
6 READA%:T=T+1:IFA%<>-1THENPOKE12288+G+T,A%:GOTO6
7 DATA0,0,0,0,0,0
10 DATA253,253,0,240,240,0,251,251
12 DATA128,255,128,255,128,255,128,255,1,255,1,255,1,255,1,255
15 DATA0,96,144,136,68,2,1,0
17 DATA255,195,165,153,153,165,195,255,1,3,4,15,20,63,72,255
20 DATA255,4,255,16,255,68,136,255,255,3,253,13,245,53,213,213
22 DATA213,214,212,216,208,224,192,128
25 DATA149,149,149,149,149,255,255,255,74,74,74,74,74,255,255,255
27 DATA165,165,165,165,165,255,255,255,82,82,82,82,82,255,255,255
30 DATA169,169,169,169,169,255,255,255
100 DATA-1
6014 FORI=0TO124+68+26:READA:POKE49152+I,A:NEXT
50000 DATA162,0,189,0,208,24,105,5,157,0,208,201,240,157,0,208,96
50010 DATA162,0,189,0,208,24,233,5,157,0,208,201,240,157,0,208,96
50020 DATA162,0,189,1,208,24,233,4,157,1,208,201,240,157,1,208,96
50030 DATA162,0,189,1,208,24,105,4,157,1,208,201,240,157,1,208,96
60014 REM*****M/C ALIEN*****
60020 DATA162,4,189,1,208,24,105,3,157,1,208,201,245,144,6,173,4,220,157,0,208
60022 DATA232,232,224,10,208,231,96
60023 DATA162,10,189,0,208,24,105,3,157,0,208,201,245,144,6,173,4,220,157,1,208
60024 DATA232,232,224,16,208,231,96
60025 DATA162,1,189,248,7,24,105,1,201,222,144,2,169,220,157,248,7,232,224,8
60026 DATA208,236,96
60027 DATA162,0,189,248,7,24,105,1,201,214,144,2,169,210,157,248,7,232,224,1
60028 DATA208,236,96
60029 DATA162,0,189,248,7,24,105,1,201,218,144,2,169,215,157,248,7,232,224,1
60030 DATA208,236,96
63000 DATA 169,15,141,24,212,169,128,141,4,212,169,7,141,5,212,169
63010 DATA 100,141,1,212,169,129,141,4,212,96
63012 FORI=0TO119:READA:POKE49500+I,A:NEXT
63015 DATA162,15,169,74,157,145,5,169,2,157,143,217,202,208,243,172,0,0,200
63017 DATA192,255,208,251,96
63018 DATA162,15,169,75,157,145,5,169,2,157,143,217,202,208,243,172,0,0,200
63019 DATA192,255,208,251,96
63020 DATA162,15,169,76.157,145,5,169,2,157,143,217,202,208,243,172,0,0,200
63021 DATA192,255,208,251,96
63022 DATA162,15,169,77,157,145,5,169,2,157,143,217,202,208,243,172,0,0,200
63023 DATA192,255,208,251,96
63024 DATA162,15,169,78,157,145,5,169,2,157,143,217,202,208,243,172,0,0,200
63025 DATA192,255,208,251,96
63030 FORI=0TO18:READA:POKE49650+I,A:NEXT
63032 DATA169,1,141,32,208,174,0,0,232,224,135,208,251,169,0,141,32,208,96
63400 POKE53280,2:POKE53281,0:POKE53272,22:PRINT"♥"
63410 PRINT"■■               I⌐ O♠LL"
63415 PRINT"■            ----------"
63420 PRINT"■              \NSTRUCTIONS"
63430 PRINT"■  IHE WALL, ♠ GAME BY _RANK IOUT          ";
63440 PRINT"   IOU ARE ¬INKY,AND YOU ARE BEING         ";
63445 PRINT"  CHASED BY JUST ABOUT EVERYONE!           ";
63450 PRINT"  IHE ♠CHOOLTEACHER,IHE WIFE,IHE HAMMER";
63455 PRINT"  EVEN YOUR MOTHER HAS A GO.               ";
63460 PRINT"   /SING A JOYSTICK IN PORT '2' GUIDE     ";
63465 PRINT"   ¬INKY THROUGH 5 WAVE'S.                 ";
63470 PRINT"   IOU MUST GET FROM THE BOTTOM OF THE   ";
63475 PRINT"  ♠CREEN TO THE CANE'S AT THE TOP.        ";
63480 PRINT"  IHERE IS A WRAP AROUND SCREEN,SO YOU ";
63485 PRINT"  CAN JUMP FROM ONE SIDE OF THE SCREEN ";
63490 PRINT"  AND APPEAR ON THE OTHER,BUT BEWARE!!!";
63495 PRINT"   IOU CAN STILL BE GOT OFF SCREEN???.   ";
63500 PRINT"■■■        ■ ⌐_¬'♠ \⌐_¬  PRESS SPACE■"
63505 GETA$:IFA$=""THEN63505
63510 IFA$=" "THEN63520
63515 GOTO63505
63520 PRINT"■■■               YOYSTICK"
63525 PRINT"  LEFT = LEFT  :  _IGHT = RIGHT            ";
63530 PRINT"  _ORWARD = CLIMB LADDER                   ";
63535 PRINT"  _IRE/BUTTON = JUMP                       ";
63540 PRINT"  ¬OINTS ARE AS FOLLOW'S                   ";
63545 PRINT"     1000♠LEVEL NUMBER                     ";
63550 PRINT"     +IONUS FOR TIME LEFT                  "
63555 PRINT"■■■        ■ ⌐_¬'♠ \⌐_¬  PRESS SPACE■"
63565 GETA$:IFA$=""THEN63565
63570 IFA$=" "THEN63600
63599 GOTO63565
63600 PRINT"■■■                 \/- ROUTINES"
63650 PRINT"   IHE MACHINE CODE ROUTINES RUN FROM    ";
63655 PRINT"   49152-49678 :♠ND CONTROL?"
63660 PRINT"   \OVE ♠PRITE'S  :  ¬HANGE ♠PRITE'S     ";
63665 PRINT"   \OVE ME    :   ¬HANGE \E              ";
63670 PRINT"   ♠OUND (ME) : "
63675 PRINT"■■ /OW ¬RESS ♠PACE IO LOAD IAME:⌐R \ IO   ■"
63780 PRINT"■        _EVEIW \NSTRUCTIONS ♠GAIN       ■"
63800 GETA$:IFA$=""THEN63800
63810 IFA$=" "THENPOKE193,2:POKE631,13:LOAD
63820 IFA$="I"THENGOTO63400
63850 GOTO63800

READY.
```



Program Listing Part 3

```
1 S1=54276:S2=54277:S3=54273:FORI=54272TO54296:POKEI,0:NEXT:POKE54296,15:GOTO630
00
2 LI=5:LE=1:AB=1:SC=0:B1=0:H=200:HI=0
3 POKE49285,225:POKE49289,223:Q=218
4 DEFFNB(P)=(PEEK(V+31)AND1):DEFFNC(R)=(PEEK(V+30)AND1)
5 POKE53272,29:Q$="␣|-▒▒▒||-▒▒▒||-▒▒▒||-▒▒▒||-▒▒▒||-"
6 C$="♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
7 D$="▒ ── "
8 E$="────────────────────────────────────────"
9 F$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒♦♦"
10 V=53248:POKEV+1,208:POKEV,240:POKEV+21,255:POKE2040,210
15 FORI=2041TO2047:POKEI,230:NEXT:GOSUB10000
28 C1=80:C2=166:C3=208:C4=140:C5=166
30 ONABGOSUB15000,16000,17000,18000,19000,50000
50 POKEV+28,255:POKEV+37,10:POKEV+38,2:POKEV+39,5
55 POKEV+29,0:POKEV+27,254
60 GG=255:FORI=3TO9STEP2:GG=GG-31:POKEV+I,GG:POKEV-1+I,0:NEXT
62 GG=255:FORI=8TO14STEP2:GG=GG-31:POKEV+I,GG:POKEV+1+I,0:NEXT
100 POKEV+15,90
500 A1=49129:A2=49203:A3=49186:A4=49152:A5=49299 :A6=49169:A7=49322
510 A8=49220:A9=49248:A0=49276:M1=49345
520 POKEV+10,40
2000 FORJ=1944TO1983:POKEJ,69:POKEJ+54272,1:I=J/10:GOSUB63103:IFSC>HITHENHI=SC
2010 POKES1,17:NEXT:B1=1984
3000 POKEV+30,0:GOTO4000
3005 IF(PEEK(V+31)AND1)<>1THEN3000
3006 POKEV+15,C1:POKEV+13,C2:POKEV+11,C3:POKEV+2,C4:POKEV+3,C5
3009 GOSUB9000:GOSUB11000
3010 IFP=103THENGOTO6010
3020 IFP=111THENGOTO7000
3030 IFP=107THENGOTO8010
3035 IFP=126THENGOTO9500
3040 IFFNC(R)=1THEN13000
3050 POKEB1,32:B1=B1-.2:IFB1<1944THEN13000
3999 GOTO3000
4000 IF(FNB(P)<1)THEN5000
4099 GOTO3005
5000 IFFNB(P)=1THENSYSA1:GOTO3000
5010 SYSA2:GOSUB12000
5015 IFPEEK(V+1)>208THEN13000
5020 GOTO3005
6010 FORT=0TO7:SYSA3:SYSA4:SYSA5:GOSUB12000:IFFNC(R)=1THEN13000
6011 NEXT
6015 FORT=7TO0STEP-1:SYSA4:SYSA2:SYSA5:GOSUB12000:IFFNC(R)=1THEN13000
6016 NEXT:GOTO3000
7000 FORT=0TO7:SYSA3:GOSUB12000:IFFNC(R)=1THEN13000
7001 NEXT
7002 IFPEEK(V+1)<75ANDPEEK(V)<40THEN14025
7005 FORT=7TO0STEP-1:SYSA2:GOSUB12000:IFFNC(R)=1THEN13000
7006 NEXT:GOTO3000
8010 FORT=0TO7:SYSA3:SYSA6:SYSA7:GOSUB12000:IFFNC(R)=1THEN13000
8011 NEXT
8015 FORT=7TO0STEP-1:SYSA6:SYSA2:GOSUB12000:IFFNC(R)=1THEN13000
8016 NEXT:GOTO3000
9000 P=PEEK(56320):IFP=123THENSYSA6:SYSA5:SYSM1
9010 IFP=119THENSYSA4:SYSA7:SYSM1
9099 RETURN
9500 IFFNB(P)=1ANDPEEK(V)<238THENSYSA3:GOSUB12000:GOSUB9600:GOTO9500
9505 GOTO3000
9600 Q=Q+1:IFQ>220THENQ=218
9605 POKE2040,Q:RETURN
10000 GG=255:FORI=3TO9STEP2:GG=GG-31:POKEV+I,GG:POKEV-1+I,0:NEXT
10010 GG=255:FORI=10TO14STEP2:GG=GG-31:POKEV+I,0:POKEV-1+I,GG:NEXT:RETURN
11000 SYSA8:SYSA9:SYSA0:A=0: RETURN
12000 SYSM1:A=A+1:IFA>2THENSYSA8:SYSA9:SYSA0:A=0: RETURN
12005 RETURN
13000 FORY=PEEK(V+1)TO220STEP3:FORTT=220TO221:POKE2040,TT:POKEV+1,Y:GOSUB63102
13005 NEXTTT,Y:POKEV+23,32:POKEV+29,32:POKEV+27,224
13010 X1=PEEK(V):Y1=PEEK(V+1)-10:POKEV+10,X1:FORI=0TOY1STEP4:POKE2045,209:POKEV+
11,I
13050 GOSUB63103:NEXT:POKEV+27,255:POKEV+23,0:LI=LI-1:IFLI<1THEN14000
13060 GOTO3
14000 POKEV+21,0:JJ=49650:KK=53281
14005 FORY=0TO200:SYSJJ:POKEKK,Y:GOSUB63102:NEXT:PRINT"␣"
14010 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒GAME  OVER"
14012 POKES2,15:POKES1,17:POKES1+2,15:FORZ=5TO100STEP3:FORZZ=10TO100STEP6
14013 POKES3,Z:POKES3,ZZ:NEXTZZ,Z:GOTO63000
14025 SC=SC+(AB*1000)+INT(B1/10):AB=AB+1:GOSUB63103:GOTO3
15000 PRINT"␣▒";:POKE53280,2:POKE53281,0
15001 FORT=0TO21:PRINTF$:NEXT
15002 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";C$
15005 PRINT"TTTTTTT♦♦♦♦♦        ♦♦♦♦♦♦        ♦♦♦♦♦♦♦♦♦♦       "
15010 PRINT"TTTTTTTT       ♦♦♦♦♦♦♦♦♦♦       ♦♦♦♦     ♦♦♦♦♦        "
15015 PRINT"TTTTTTTT♦♦♦♦♦        ♦♦♦♦♦♦        ♦♦♦♦♦♦♦♦♦♦        "
15050 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";Q$
15055 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";Q$
15060 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";Q$
15065 PRINTD$
15070 PRINT"▒▒▒▒":PRINTTAB(33);SC
15075 PRINT"▒▒▒":PRINTTAB(33);LI
15080 PRINT"▒▒▒":PRINTTAB(33);LE
15082 IFSC>HITHENHI=SC
15085 PRINT"▒▒▒":PRINTTAB(33);HI
15090 POKEV+15,90
15098 GOSUB60000
15099 RETURN
16000 POKE53280,8:POKE53281,11:PRINT"␣▒":POKE49285,229:POKE49289,227
16005 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";C$
16010 PRINT"TTTTTTT♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
16015 PRINT"TTTTTTT♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
16020 PRINT"TTTTTTT♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
16030 PRINT"▒";:FORT=0TO21:PRINTF$:NEXT
16040 :PRINTD$:PRINTTAB(26)"▒▒▒▒▒";Q$
16045 PRINTTAB(4);Q$
16050 PRINTTAB(10);Q$
16060 C1=70:C2=166:C3=200:C4=130:C5=170:LE=2
16075 PRINT"▒▒▒▒":PRINTTAB(33);SC
16076 PRINT"▒▒▒":PRINTTAB(33);LI
16080 PRINT"▒▒▒":PRINTTAB(33);LE
16082 IFSC>HITHENHI=SC
16085 PRINT"▒▒▒":PRINTTAB(33);HI
16099 GOSUB60000:RETURN
17000 POKE53280,5:POKE53281,3:PRINT"␣":POKE49285,232:POKE49289,230:LE=3
17002 FORT=0TO21:PRINT"▒                                         ":NEXT
17005 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";E$
17010 PRINT"TTTTTTT    ───     ───     ───   "
17015 PRINT"TTTTTTT──────   ───     ──     ──"
17020 PRINT"TTTTTTT─────      ──     ───────     ─":PRINTD$
17025 PRINTTAB(18)"▒▒▒";Q$:PRINTTAB(4)"␣";Q$
17035 PRINTTAB(22)"|-▒▒▒||-▒▒▒||-▒▒▒||-▒▒▒||-"
17075 PRINT"▒▒▒▒":PRINTTAB(33);SC
17076 PRINT"▒▒▒":PRINTTAB(33);LI
17080 PRINT"▒▒▒":PRINTTAB(33);LE
17082 IFSC>HITHENHI=SC
17083 PRINT"▒▒▒":PRINTTAB(33);HI
17085 C1=70:C2=166:C3=200:C4=50:C5=180:LE=2
17099 GOSUB60000:RETURN
18000 PRINT"␣▒";:POKE53280,5:POKE53281,0
18001 FORT=0TO21:PRINTF$:NEXT
18002 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";C$
18005 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";Q$;"▒▒▒";Q$;"▒▒";Q$
18010 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒      ♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
18015 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒   ♦♦":PRINT"▒▒▒▒▒▒▒▒▒▒▒▒  ♦␣♦♦♦♦♦▒♦♦♦♦♦♦♦♦▒▒▒♦♦♦♦♦♦♦♦♦"
18020 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒                 "
18025 PRINT"▒▒▒▒▒▒▒▒";Q$;"▒▒▒";Q$
18030 POKE49285,236:POKE49289,233:LE=4
18040 C1=62:C2=120:C3=142:C4=115:C5=140
18070 PRINT"▒▒▒▒":PRINTTAB(33);SC
18075 PRINT"▒▒▒":PRINTTAB(33);LI
18080 PRINT"▒▒▒":PRINTTAB(33);LE
18090 IFSC>HITHENHI=SC
18095 PRINT"▒▒▒":PRINTTAB(33);HI
18099 GOSUB60000:RETURN
19000 PRINT"␣▒":POKE53280,6:POKE53281,0
19001 FORT=0TO20:PRINTF$:NEXT:PRINTC$,D$
19005 PRINT"▒▒▒▒▒♦♦          ␣♦♦♦♦▒▒                 ♦▒♦♦♦♦♦♦"
19010 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒♦♦♦          ▒♦♦♦       ▒▒♦♦♦  T♦♦♦♦♦"
19050 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";Q$;"▒▒▒";Q$:PRINTTAB(26)"TTTTTTTTTTT";Q$;"▒▒";Q$
19069 POKE49285,239:POKE49289,236
19070 PRINT"▒▒▒▒":PRINTTAB(33);SC
19075 PRINT"▒▒▒":PRINTTAB(33);LI
19080 PRINT"▒▒▒":PRINTTAB(33);LE
19082 IFSC>HITHENHI=SC
19085 PRINT"▒▒▒":PRINTTAB(33);HI
19086 C1=162:C2=80:C3=189:C4=165:C5=60:LE=5
19090 GOSUB60000
19099 RETURN
50000 AB=1:GOTO3
50010 REM
50020 REM
50030 REM
50040 REM
60000 PRINT"▒▒▒":PRINTTAB(33)"SCORE":PRINT"▒":PRINTTAB(33)"────────"
60010 PRINTTAB(33)"LIVES":PRINT"▒":PRINTTAB(33)"────────"
60020 PRINTTAB(33)"LEVEL":PRINT"▒":PRINTTAB(33)"────────"
60025 PRINTTAB(33)"HI-SC"
60030 IFSC>HITHENHI=SC
60099 RETURN
63000 V=53248:POKEV+21,255:POKEV+29,255:POKEV+23,255:POKEV+28,0:POKEV+39,1
63002 J=202:FORI=2040TO2047:POKEI,J:J=J+1:NEXT:POKEV+1,100:POKEV+13,100
63005 POKE53280,0:POKE53281,0:PRINT"␣"
63010 FORI=3TO13STEP2:POKEV+I,100:POKEV-1+I,155:NEXT:POKEV,0:POKEV+16,64:POKEV+1
4,65
63013 X1=165:X2=165:X3=165:X4=165:X5=165:XX=4:POKEV+16,64:X7=71:X8=155:Y1=0
63015 FORX=165TO76STEP-3:XX=XX+1:POKEV,XX:POKEV+2,X:X2=X2-1.5:POKEV+4,X2:POKEV+6
,X3
63025 X4=X4+1.5:POKEV+8,X4:X5=X5+3:POKEV+10,X5:POKEV+16,64:X7=X7-1:POKEV+12,X7
63030 GOSUB63100:POKEV+14,X8:Y1=Y1+5:POKEV+15,Y1:NEXT
63050 PRINT"▒▒▒▒▒▒▒BY FRANK TOUT PRESS FIRE BUTTON FOR GAME"
63090 IFPEEK(56320)=111THENPOKEV+23,0:POKEV+29,0:POKEV+21,0:POKEV+16,0: GOTO2
63099 LI=5:GOSUB63200:GOTO63090
63100 POKES1,32:POKES2,15:POKES1+2,60:POKES3,Y1:POKES1,33:RETURN
63101 POKES1,32:POKES2,15:POKES1+2,60:POKES3,I/10:POKES1,33:RETURN
63102 POKES1,128:POKES2,15:POKES3,Y:POKES1,129:RETURN
63103 POKES1,32:POKES2,15:POKES3,I:POKES1,33:RETURN
63200 S1=54276:S2=54277:S3=54273:POKE54296,15:BB=250
63201 POKES1,34:POKES2,15:READA:POKES3,A:POKES1,33
63202 FORT=0TO150:NEXT:IFPEEK(56320)=111THEN63090
63203 IFA<>000THEN63201
63210 RESTORE:RETURN
63300 DATA10,20,30,40,40,20,10,30,20,10,30,40,50,50,40,30,20,10,10,40,60,70
63310 DATA70,60,70,80,40,20,10,20,20,10,5,10,20,40,80,120,120,80,40,20,10
63320 DATA60,60,60,70,70,70,80,80,80,70,80,70,60,50,40,20,30,40,50
63330 DATA20,30,40,50,30,40,50,30,40,50,30,40,50,60,70,80,70,60,50,40,60,40,30
63340 DATA20,20,20,15,10,5,10,5,10
63350 DATA5,10,10,5,10,10,5,5,6,4,5,10,10,9,8,9,5,10,10,10
63360 DATA5,10,10,5,10,10,5,5,6,4,5,10,10,9,8,9,5,10,10,10
63370 DATA5,10,10,5,10,10,5,5,6,4,5,10,10,9,8,9,5,10,10,10,20,30,40,50
63380 DATA50,50,40,50,50,40,50,50,40,50,50,70,70,70,60,60,60,40,30,20,10,5,6
63385 DATA50,50,40,50,50,40,50,50,40,30,50,40,30,20,40,30,10,5,10,5,4,6,10
63400 DATA10,10,10,20,20,20,20,000

READY.
```

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema.

Most of them comply with the rules contained in the British Code of Advertising Practice.

But some of them break the rules and warrant your complaints.

If you're not sure about which ones they are, however, drop us a line and we'll send you an abridged copy of the Advertising Code.

Then, if an advertisement bothers you, you'll be justified in bothering us.

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.

COMPUTAPE'S DISCOUNT SOFTWARE

This is but a sample of the range of games available from our catalogue. Take a **'LOOK'** at our keen prices.

Commodore 64		RSP	Our Price
Arabian Nights	Interceptor	£7.00	£5.80
Automania	Mikro-Gen	£7.95	£6.80
Bear Bovver	Artic	£6.95	£5.90
Black Hawk	Creative Sparks	£7.95	£6.90
Boulder Dash	Statesoft	£8.95	£7.80
Bristles	Statesoft	£8.95	£7.80
Catacombs	Anirog	£7.95	£6.75
Chuckie Egg	A & F Software	£7.90	£6.50
Cybotron	Anirog	£7.95	£6.80
Daley Thompsons Decathlon	Ocean	£7.90	£6.50
Decathlon	Activision	£9.99	£8.70
Encounters	Novagen	£9.95	£8.50
Evil Dead (The)	Palace	£6.99	£5.90
Flip and Flop	Statesoft	£8.95	£7.75
Football Manager	Addictive Software	£7.95	£6.50
Glug Glug	C.R.L.	£7.95	£6.75
"Hareraiser" Prelude	Haresoft	£8.95	£7.50
House of Usher	Anirog	£6.95	£5.80
"Hulk" The	A/International	£9.95	£8.50
Jack and the Beanstalk	Thor	£6.95	£5.80
Micro Olympics	Database	£5.95	£4.80
Monty Mole	Gremlin Graphics	£7.95	£6.70
Mystic Mansion	U.S. Gold	£8.95	£7.90
Panic	Interceptor	£7.00	£5.80
Petch	Anirog	£6.95	£5.70
Pitfall	Activision	£9.99	£8.70
Potty Pigeon	Gremlin Graphics	£7.95	£6.70
Stunt Bike (The)	Ocean	£6.90	£5.75
Trashman	Quicksilva	£7.95	£6.80
Twin Kingdom Valley	Bug Byte	£9.50	£7.95
Vortex Raider	Interceptor	£7.00	£5.90
Zaxxon	U.S. Gold	£9.95	£8.50

SPECIAL OFFERS for Christmas and the New Year, so hurry and order NOW!

Astro Chase	Statesoft	£8.95	£7.30
Beach Head	U.S. Gold	£9.95	£8.10
Flight Path 737	Anirog	£7.95	£6.30
Gilligans Gold	Ocean	£6.90	£5.40
Hunchback	Ocean	£6.90	£5.40
Psytron	Beyond Software	£8.95	£7.20
Quo Vadis	Softek	£9.95	£8.40
Valhalla	Legend	£14.95	£10.30

To order any of the above tapes, please write giving details of the games, enclosing a Cheque/P.O. made payable to: **COMPUTAPE** and post to: **Dept. Y129, 27 Coombe Road, Southminster, Essex CM0 7AH. (Tel: 0621 772589).** Prices inclusive of postage, but please send SAE for catalogue only. Lists also available for the Spectrum, BBC, Dragon, Atari and Tandy Computers.

Braingames

The tough nuts to crack!



FLAME ISLAND

For the Commodore 64

Disk £12.95
Cassette £8.95

BRAINGAMES Amplicon Group
Richmond Road Brighton East Sussex BN2 3RL
Tel: Brighton (0273) 608331 Telex: 877470 AMPCON G
Braingames is a division of Amplicon Micro Systems Limited

Braingames

The tough nuts to crack!



CASTLE FEAR

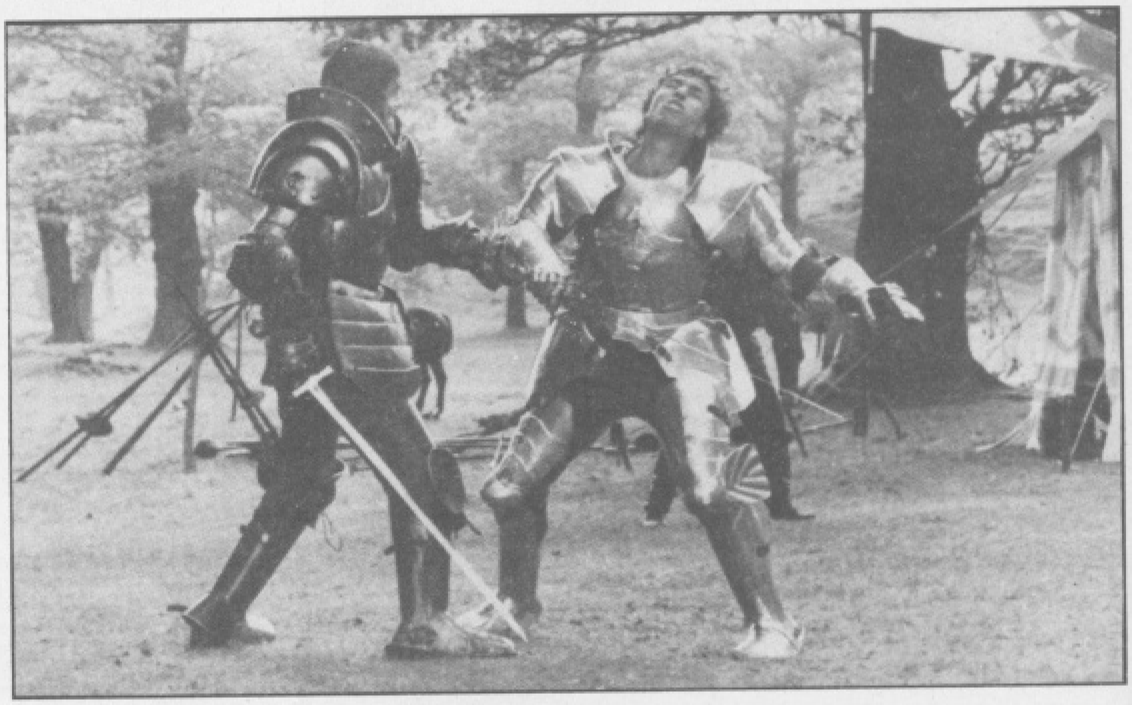
For the Commodore 64

Disk £12.95
Cassette £8.95

BRAINGAMES Amplicon Group
Richmond Road Brighton East Sussex BN2 3RL
Tel: Brighton (0273) 608331 Telex: 877470-AMPCON G
Braingames is a division of Amplicon Micro Systems Limited

Runecaster returns with guidance on how to keep your cool when lost in a maze.

SENSE OF ADVENTURE



LAST MONTH WE LOOKED AT MOVING into adventure games, with how to map what we find. This is a really vital part of adventure gaming — unless you are one of those people with eidetic (photographic) memory.

To recap, look at figure 1. This is the sort of map you would expect to have after exploring the first few locations of a new game. Each box represents a discrete location and the lines between them show how these are linked together. Arrows signify the direction you may take to reach the location. Only one arrow would mean movement in only one direction and the cross-bars indicate that no exit (at the moment!) exists in that direction. The loop to the east of the 'steep river bank' shows that if you continue going east from this location — you end up at the same position! This technique is often used at the 'edge' of an adventure's known world.

Each box is labelled so that you will be able to relate it's position to the description seen on the screen. Items found — and if not obvious how found — are also recorded.

All this is fairly straightforward, providing you have a large enough piece of paper and keep your head — sometimes literally! Now look at figure 2. Ouch, what a mess. One slip of the pencil and you could land yourself in real trouble! What's worse, nearly all the paths are one way only. AND, what if all four locations had the same description (I have labelled them ABCD just for the purpose of this article!)?

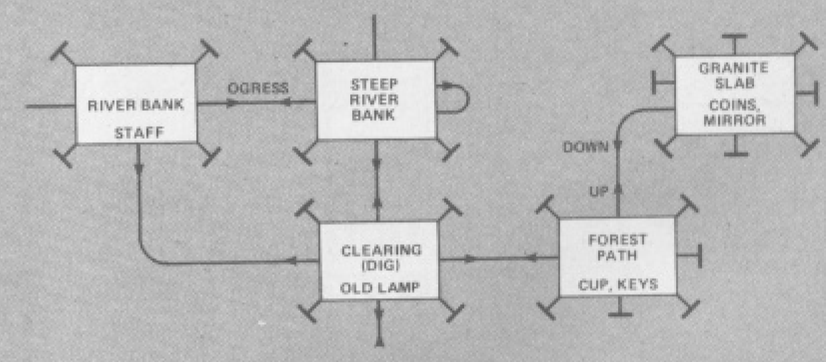


Figure 1

Just amazing

As you've surely guessed, you are in a form of maze. Directions no longer necessarily follow the logic used elsewhere in the game. Going west, having just moved east, is not likely to get back to where you started! Figure 2 shows a four location maze but rarely will you find one with so few 'rooms'. You can often find your way out by just hitting the direction keys at random, but this is seldom the whole story and is unlikely to help you solve the adventure. Either you have wasted a number of valuable turns during which your food or lamp is running out — and of course the program is structured so that it is then impossible to reach more food/oil etc. Or hidden somewhere in the heart of the maze, is something vital to your wellbeing or vital to a successful conclusion in the future!

You have got to face up to the problem of mapping the maze. Mazes seem to be an accepted part of most adventure games. Some people would say that they are unnecessary as they tend to be similar puzzles. But as the cunning of the programmer grows, so too does the type of maze they produce and we the players have got to look for clues to pinpoint what varient we are being called to solve!

The 'original' type of maze is not so difficult to overcome and the basic procedure used to solve it, may well form the core of how you approach its many varients. Firstly, you must use a different form of 'mapping', something like figure 2 would produce a very messy and unreliable map of the area altogether.

Obviously you will not always know in advance that you are about to enter a maze! So this immediately hammers home the need to make frequent SAVEings of your progress. Nearly all of the better adventure games include the facility to SAVE your character's position up to wherever you have reached at that time. Adventure games are very rarely written with the intention of a player sitting down and solving them in one sitting! They are meant to provide you with days or weeks of playing time before you reach a solution.

So use the SAVE facility fairly often — do not overwrite a previous SAVE *all* the time, otherwise you may find that the objects you now hold are the objects you need — one of those previous SAVEs may save you a lot of time! Using this form of recursive approach is **not** cheating, it is part of the system by which you learn your way around.

Right . . . you may find yourself in a maze — so QUIT! ReLOAD your last SAVE and approach the maze carefully! Make sure you are carrying as many objects as the game permits — now when you enter the maze, drop something at location 'A'

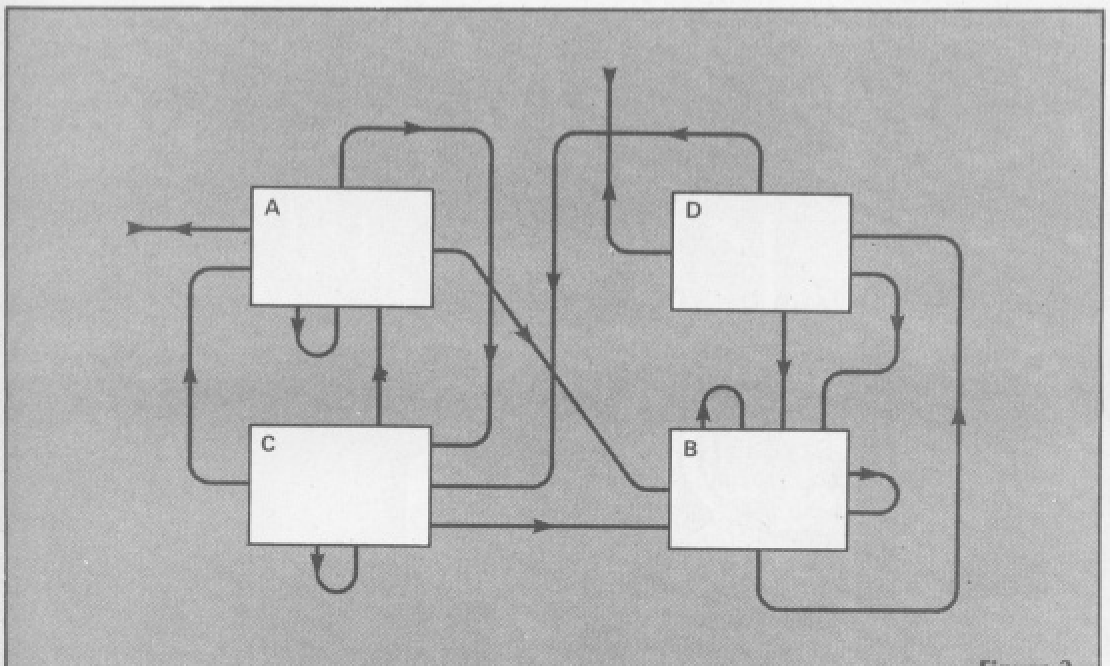


Figure 2

(we'll just use fig 2 as our maze) say 'a staff'. Instead of following our previous mapping technique we'll draw a separate, and unconnected, box for each location.

Although the descriptions are the same, location 'A' will also have 'a staff' lying around! Move in any direction — if you moved south you will return to location 'A' ...and will see the staff, but if you go east you will come to another seemingly identical place — but without the staff! So drop something else . . . a cup. East now, and you are back to the cup, west, and yet another location . . . drop some keys. North . . . lo and behold, back to where you dropped the staff.

something extra, then it is still unique — mind you, this is a pretty dirty trick and certainly gets the adrenalin pumping, until you realise what is happening!

Once you have mapped your maze, you can either retrace your steps and pick everything up or QUIT, reLOAD and travel through carrying your precious (you hope) artifacts with you — just don't forget where you are!

Although this kind of maze is disappearing in favour of more complex nasties — it still points the way to how to solve most if not all mazes. You have **got** to find a way to **uniquely** define each location.

Perhaps your maze is a mixture of the two above types — ouch? Just keep your cool and hope that there is a logical solution. "You are on a path in a forest", "You are on a path in a forest", "You are on a path in a forest" — how about EXAMINE TREES or EXAMINE PATH — you may well find a clue at one or all locations!

Another devious one is for your location to alter something you are already carrying — perhaps your sword tarnishes or that magic mirror you have been holding onto, wondering what use it is — is showing different reflections as you move around the maze?!

The ploys and puzzles within mazes have by no means been wrung dry and a well wrought, logical adventure maze can still give tremendous satisfaction...when solved!

Thorny subject

Much has been written about tape copy programs. We all know that one of their primary uses is to cheat the software houses out of their rightful dues. All of them warn about this usage — but still, thousands of people "just make a copy for a friend"!

It is of course nearly impossible to stop. Software houses can tackle the problem in three ways: (1) endeavour to make their programs 'uncopyable' (2) the bother or (3) increase the price to make up for losses.

If you want a good product it is almost certain that it was costly to produce. Often, teams of programmers have worked on different aspects — music, graphics, etc — for months.

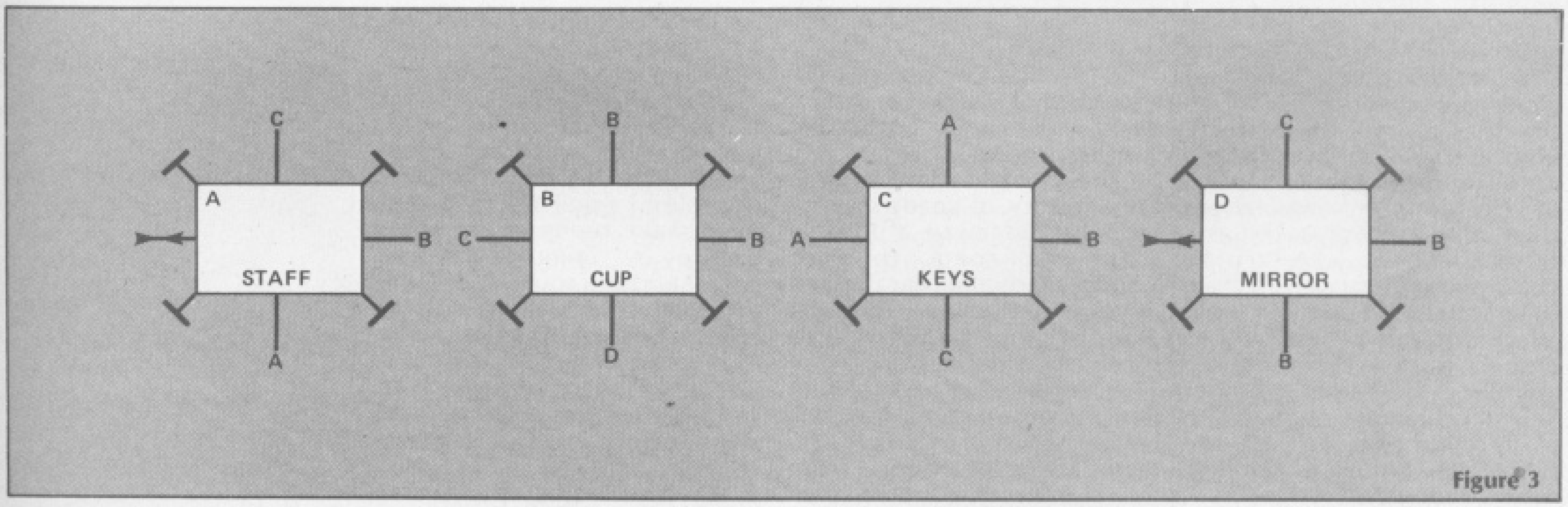


Figure 3

To be sure you have mapped the maze **you must** try every possible direction from each location. Having done this you will end up with a map like that in figure 3. You **must** try every direction; programmers are quite devious enough to make the only acceptable 'NW' move in the whole game, the way out of the maze to somewhere vital!

To solve a maze such as this you must be able to carry enough objects to drop in the various locations! If you think about it, you will realise that it is possible to solve a maze with **one less** object than there are locations. If only one place does **not** have

Take care

Sometimes the solution to a maze is staring you in the face — literally. Read the descriptions of all locations **very** carefully. "You are on a raft in a storm swept sea there are sharks all around you" looks very similar to "You are on a raft in a storm swept sea, there are sharks all around you". It is easy enough to spot the difference on this printed page but when those two descriptions follow one another on your screen — will you always notice the extra comma? Or a space, or a full stop at the end of one of them?

Having said my piece, let us look at the other side of the coin: (1) it is useful to have **your own** back-up, in case of catastrophe and (2) although many adventure games are now on 'Turbo Load' (and it's many variants), some take a long time to LOAD...

I recently got a copy of HYPER-SAVE 64 for my CBM 64 — produced by DOSOFT of Blackpool. This utility enables a Hyperspeed copy of a standard LOAD cassette to be made. Not all tapes will be accepted but I would certainly recommend having one to hand, on the off chance! Impatient — that's me!

SOFTWARE SPOTLIGHT



Need some help in compiling your Christmas list? Let our reviewers guide you through another jungle of Commodore software.

House of Usher
★ ★ ★
Anirog
£6.95
CBM 64 + joystick

QUESTION: COULD YOU BE responsible for the fall of the House of Usher? Answer: quite simply, yes. But it will take some doing. It's not that there's anything desperately new to get used to or, indeed, anything desperately difficult about it. No, it's just that there is a lot to get through. The concept of the game, however, is relatively novel although perhaps a little disappointing in terms of excitement value. But on with the game. The warped and twisted minds of generations of mad Ushers have devised a variety of tormenting challenges. On entering the ancestral home you are given a choice of nine rooms to enter in each of which there is a game to play. As stand alone games go it's fair to say that they are not up to much. Suffice to say, although they are all different, there is a touch of the 'donkey kongs' about them all. However, taken as a whole they are quite challenging because you have to successfully negotiate them all before you are allowed to enter a further two rooms to complete the game. Whether you live long enough to mastermind the Fall of the House of Usher is another matter though.

K.M.

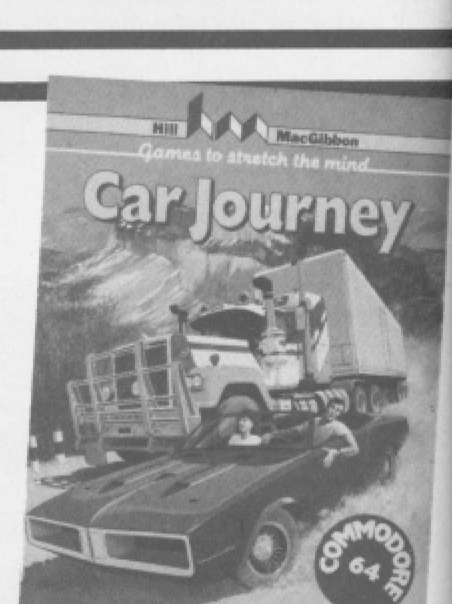
Car Journey
★ ★ ★ ★
Fourways Software & Hall MacGibbon
£9.95
CBM 64

THIS IS A BOOKLET/CASSETTE educational package which deals with sundry aspects of road transport.

The booklet starts with a brief history of roads starting with the Ridgeway, a possible stone-age road, through Roman roads and Macadam to modern motorways. The next part of the booklet deals succinctly with the workings of a car. Further sections deal with petrol consumption and the layout of roads in Britain. An interesting diversion from technicalities is a short extract from the "Wind in the willows" (Toad rather enjoys motoring... if you didn't know), a footnote suggesting that the child reads the rest of the book. Finally there are several games to play on the theme of car journeys. The booklet is very well thought out and presented.

The program supplied in the package endeavours to put into practice what has been learnt in the booklet. You play the role of a delivery man and must make money from your service by using your van, petrol and the routes available efficiently. Calculations have to be made throughout the game. The graphics are simple but effective and can be used in further exercises suggested in the booklet as well as the main routine.

Overall the program appeared to be the weakest



section of the package, but used as a whole would serve as a good addition to any classroom.

M.E.W.

Terrorist
★ ★ ★ ★
Virgin Games
£7.95
CBM 64 + joystick

THE CITY HAS GOT A SEVERE CASE OF the terrorist attacks and Red Leader, the head of the anti-terrorist squad has been given the task of putting a stop to all the mayhem. First choose your rank: private, sergeant, captain or colonel and then study the map you have been given. Don't take too long because the longer you take the more points you lose. Then it is off to the trouble spot. To transport you around the attack zone at speed, a helicopter has been placed at your disposal. You also have a police car and a boat to use in uncovering the roads and the rivers in the area. You transfer from one to another simply by pressing the allocated function keys. You can use the helicopter to transport both the car and the boat to other parts of the attack zone. Points are scored for uncovering the roads, rivers and the attack locations and points are lost for going in the wrong direction or not moving at all. But watch your fuel levels closely. Fuel can be transferred to the boat and the car from the helicopter when it runs out but the helicopter must take on more fuel from the fuel dump before it crashes. The helicopter has three 'lives'. All locations under attack are revealed by a shell burst and you must rush to them immediately. When you get there the screen changes to show the area in more detail. Silhouettes of the terrorists will appear but you must get in the first shot. At the end of the battle the casualties are counted and the location declared either safe or in the enemy's hands. So do your good deed for the day. A great game.

K.M.

Borzak
★
Channel 8
£6.95
CBM 64



ISN'T IT NICE TO KNOW THAT at the end of a long and tedious day you can relax and rest assured that Borzak, the amazing bug-eyed beastie from Betelgeuse has been banished from whence he came? Apart from the fact that the joystick option was unresponsive, an irritating midemeanour to be sure, about the best part of this game was the 'game over' screen which finally put Borzak out of his agony. The object of the game is to get Borzak back to his "amazing intergalactic craft" avoiding a variety of obstacles emerging from the right of the screen as it scrolls from right to left. You'll be glad to know that there is a pause facility although I managed to avoid temptation. Enough said.

K.M.

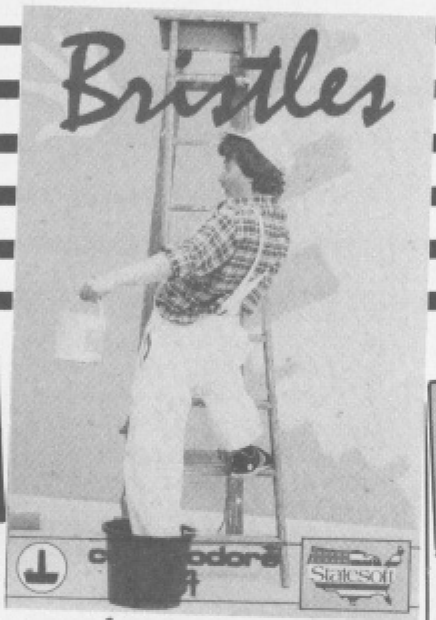
Ghouls
★ ★ ★
Micropower
£6.95
CBM 64 + joystick (optional)

THEY SAY THAT OLD GHOSTS die hard, well something like that! It's true, I am having real difficulty with this one. Ghouls by Micropower is a vexing game. The concept of the game is to retrieve the Power Jewels from a rather nasty mansion. Within this delectable abode you will find poison tipped spikes, moving floors, spiders and the Ghoul.

Your man, who resembles a distant relative of a Pacman with legs, has to traverse four floors of this mansion to get to the Power Jewels. The only aids you have are the occasionaly spring to step on and the odd Power Jewel to eat. Upon digesting this appetiser the Ghoul, who is slowly descending on you, disappears for a few seconds allowing you to continue your task unhampered.

If you do die, and I can assure you it's very likely, the Ghoul will grin with pleasure. But you soon return to try and complete the level and move on to harder screens. When your little Pacman completes the fourth and final screen the Ghoul disintegrates before your eyes and Pacman jumps up and down.

The game also contains a facility to turn sound off, which includes all the sound effects. It's a very fast game as you have a time limit to complete each screen in. There is a high-score table and a pause button as well as the fast loading system!



Bristles
★ ★ ★ ★
Statesoft
£8.95
CBM 64 + joystick

AT LAST A SOCIABLE computer game and a good one into the bargain. Bristles has a four player option and gives each player the choice of playing two painters. And that's the object of the game...simply to paint all the rooms in a variety of houses as quickly as possible. There are six skill levels to progress through and eight houses in each skill level, so it's not exactly an easy task. It's not just a high score you're after either. On each skill level there is a hidden message which will appear in part after every house you complete the painting contract on. The ultimate challenge is to discover all the messages. What, no obstacles? Well not quite. In fact there are quite a few. Each player starts the game with ten paint brushes and gets an extra two for each house painted but also loses one for each mistake made. There are lifts to help you get from floor to floor but unless you time it right they will splatter you into the basement. There are also three ladders located in the lift shafts in the basement but, again, unless you are quick, the the lifts will splatter you. But the major obstacles are the flying half pints of paint which you have to jump over or duck under and the dumb buckets. The bucket chucker is highly intelligent and will chase you from room to room. And there's more. Brenda the Brat is intent on making hand prints all over your freshly painted walls and the only way to stop her is by giving her a sweetie. There are also the steam pipes to watch out for which just happen to occassionally protrude through the floors of some rooms. Oh, I almost forgot: sometimes you are painting with clear varnish and at other times you are painting in the dark. Gone off painting and decorating a bit? Well I'm not surprised but definitely give it a go before you give up completely.

K.M.



Ant Attack
★ ★ ★ ★
Quicksilva
£8.95
CBM 64 + Joystick

THIS GAME HAVING BEEN A great success on the Spectrum has now been converted for the 64. At the time of it's original release, there was great acclaim for the unusual use of the 3D effects. To my mind the effects on the 64 are also very good albeit not quite as impressive as the Spectrum version.

Notwithstanding the absurd spiel on the cassette inlay, the plot is tolerably simple. You control a small figure and attempt to locate unfortunates scattered about the deserted city of Antescher. The buildings of the city are represented as blocks and you get a nice perspective view of part of the city. As you move about, the scene scrolls diagonally showing more of the area. By pressing any of the function buttons, you can choose any one of four views of the area. The main problem is that the city is occupied with large quantities of giant ants which have this irritating habit of killing our hero. To evade death, you can either try to out run them, climb up a building or throw one of your limited supply of grenades.

When you encounter one of the lost souls, they make a suitably banal comment and, if you don't run too fast, follow you to safety. You must be careful, however, to protect both yourself and the rescued victim.

The graphics are superbly simple comprising of grey blocks with different shades of grey to give perspective. The movement of the little man is accurate in as much that you can climb over, through and around the buildings. The animation of all figures is superb. The use of sound, on the other hand, was weak.

Overall, whilst the visual effect is faultless, I found the play rather boring. After the first half hour, I found myself dearly wishing that something new would happen.

A.E.W.

SOFTWARE SPOTLIGHT



High Noon

★ ★ ★ ★
Ocean Software
£7.90
CBM 64 + joystick

SO YOU FANCY YOURSELF AS a bit of a cowboy do you? Care to live out your wild west fantasies? Anyway, here's your chance. High Noon leans heavily on this macho world to provide a neat little game plot. Needless to say you are the good guy in the scenario and, armed with your pistols, it is your responsibility to keep the peace in a not-so-quiet frontier town. You are up against a desperate gang of bandits intent on stealing the gold from the bank and the girls from the saloon. They'll come on horseback and on foot and armed with dynamite or just their six guns but, whatever it is it is bound to lead to a shoot out on main street. To prevent the loss of your bonus points you have to keep the girls and the loot safe so shoot first and ask questions later and remember, they won't shoot when there is a chance of shooting each other...the cowards. Of course there are a number of screens to progress through each of which is increasingly difficult. Eventually, if you're good enough you will track them down to their hide-out in the hole in the wall when you will be faced with the final showdown. That's always assuming you escape the clutches of the overworked undertaker, Riga Mortis.

K.M.



Secret Agent

★ ★ ★ ★
Fiveways Software & Hill MacGibbon Ltd.
£9.75
CBM 64

THIS IS AN EDUCATIONAL game with a difference. It comes as a package of cassette and booklet. The booklet deals with real-life spies and those from fiction. After reading about spies, the reader is given activities to try, either writing a spy story or doing further research. To be a successful spy you will need to understand about codes and spy kits. Similarly, a good grasp of Europe and it's cities is necessary. The booklet gives information about all these aspects and suggests extra activities. Having done the background work from the booklet, the game puts theory into practice.

Your aim in the game is to capture an enemy agent who is travelling about Europe killing off your resident agents. You have secondary aims in that you try to operate as fast as possible and to keep costs down. Fail to keep to your budget and to operate speedily and you will end up pen-pushing behind a desk. A highly unworthy end for a secret agent!

During the game your agents will send intelligence reports giving clues as to the whereabouts of the enemy. Some of these reports have the added difficulty of being in code. You can get help from headquarters to de-code the message but it costs extra money, as do messages from informants. When you think you know where the enemy is, you can dial up a time table of flights and trains out of town and to other cities. You can then travel to a city where you think he is and, with luck, capture him.

This is a very good package in that it brings together all aspects of education. These include geography, history, the need to perform simple decoding and even the ability to use the 24 hour clock.

In all, this is an excellent package which can be used as a fun way to spend time at home, or can be usefully expanded to a project for use at school.

M.E.W.

Cybertron Mission

★ ★ ★
Micropower
£6.95
CBM 64 + Joystick (optional)

THIS IS YET ANOTHER COPY OF YET another successful game for the Atari. This is a copy of a game called Shamus in which you have to run through rooms searching for treasure while eliminating clones, drones or as they are called in this game, Spinners, Clones, Cyberdroids and a ghost of a player past. The difference between this one and other copies is the task you have to perform.

In this game the player is told what treasure has to be retrieved and put in the safe. On the first level it is one treasure, on the next it is two and so on. I must point out that just before you enter a level there is this superb sound effect. It only lasts for a second or so but it's great. Anyway, there are 16 rooms on each level to explore while looking for the treasure, an indicator in the top right-hand corner indicates which room and what level you are on.

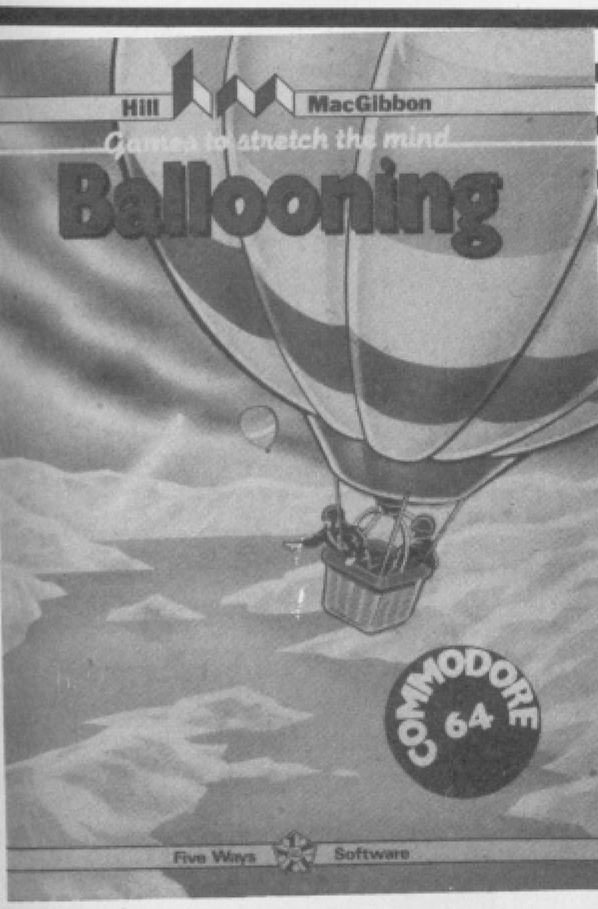
I mentioned earlier the existence of a ghost. He cannot be killed but he can be stunned. Obviously as you travel through each level the spirit, when stunned, isn't stationary for a long a period. The game has reasonable graphics with good sound accompaniment. It has a pause facility and a high score table. I would have preferred the treasures and the safe in a different colour as they are occasionally hard to find.

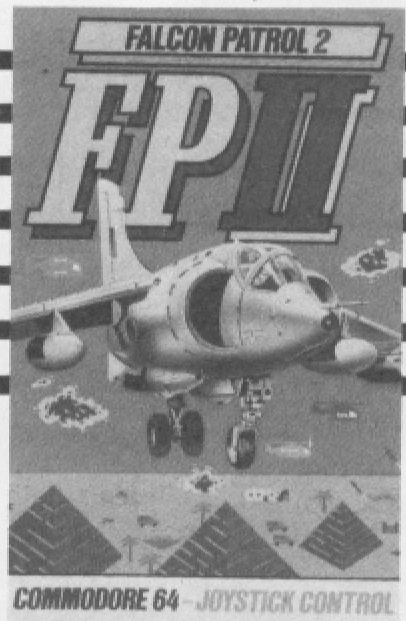
S.L.F.P.

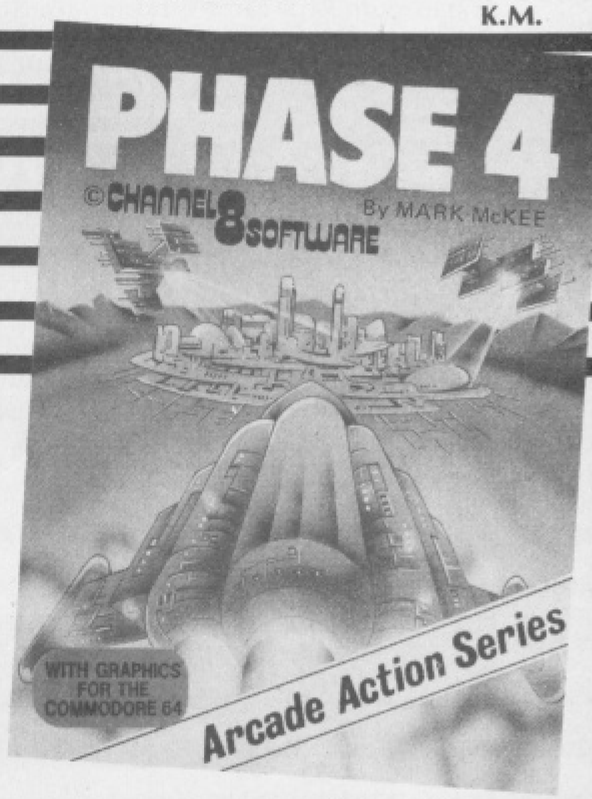
Falcon Patrol 2
★ ★ ★ ★
Virgin Games
£7.95
CBM 64 + joystick

YOU HAVE GAINED THE upper hand in the war with your deadly enemy and it seems to be your VTOL that has done the trick. But now the enemy is about to launch a last ditch attack to win the war and it is up to you to prevent them from succeeding. For the battle, your VTOL has been armed with both air-to-air and air-to-ground missiles. Although your VTOL is nippy and highly manoeuvreable, it requires careful handling. You can't just turn around and chase the enemy, you have to slow down first. And the enemy is no pushover either. Although they have no VTOLs, they have three types of helicopters: transports which drop flak batteries and radar jammers, gunships which can shoot you down, and solos designed to lure you into making mistakes. A radar display at the bottom of the screen will indicate the enemy's position unless radar jammers have been dropped. Fully fuelled and fully armed the VTOL carries 100 missiles but you are bound to need to take on more weapons and fuel during the course of the battle. You do this by landing on the strategically located launch pads which remain serviceable until 75% are destroyed. But there are dangers in this. You are vulnerable and the enemy knows it. So beware otherwise you'll never make it through the 16 levels.

K.M.







Magic Micro Mission
★ ★
Quicksilva
£7.95
CBM 64 + Joystick

THIS GAME OFFERS TWO GAMES OF different but unexceptional styles. The first game involves pseudo 3D movement effects to simulate the view forward from the cockpit of a spaceship. The idea of the game is to shoot approaching nasties before they reach you. You have a cross-hair sight to assist your aim. The graphics and use of colour were very nice but the actual action was appallingly unexciting.

The second game is a maze type and, much to my surprise, was worse than part one. You control a robot and must move about a maze endeavouring to catch rampant ROMs (Read Only Memories). Your movement is impeded by patrolling resistors and RAM chips. If you get into trouble you can disable the opposition but you loose 5000 points. Since scoring was low, I didn't have this option available very often. Graphically this segment was only just average.

Overall, I found this package weak and really not any better than average.

A.E.W.

Ballooning
★ ★ ★
Five Ways Software & Hill MacGibbon
£9.95
CBM 64

USING THE WORDS OF THE accompanying booklet..."Ballooning is an accurate simulation of a hot air balloon". In fact it was so accurate that the ascent took so long that it was very easy to fall asleep, thus missing the whole point of the game (you should be observing the dials giving your altitude and fuel etc.). You start off at flying school learning how to fly and land your balloon. Following this, when you think you are ready, you take your flying test and hopefully gain your wings. Should you wish to do a little extra work, you can keep a record of your altitude during your flights and try to make a duplicate flight. The booklet accompanying the cassette suggests alternative games which you can play using the package.

The booklet covers many aspects of balloons and ballooning starting with their history through to the scientific principles of their flight. Suggestions for further research are provided so that the topic can be stretched as far as you want. An important aspect of flying is the ability to read maps and estimate the effects of wind. Again this aspect is discussed in the booklet.

The display gives a side view of your balloon moving across the terrain. Along the bottom of the screen are instrument dials showing altitude, rate of climb, fuel and temperature. Overall a reasonable package although the program seemed a little slow and tedious.

M.E.W.

Phase 4
★ ★ ★
Channel 8
£6.95
CBM + joystick

NO CONCESSIONS TO THE FAINT-hearted with this zapping game. It's straight in at the deep end with the kind of difficulty levels that killed video gaming as a pub sport. No sooner had you put your 10p in the slot when the game was over. There are ten or so different types of alien craft to zap most of which are firing at you at the same time and all of which have a different value in terms of points. Take the first screen for example. It looks like you're defending the pyramids and not only do the aliens flit across the screen casually dropping tracers but there's one that homes in on you as well. And, with such a highly coloured background it's hard to tell where anything is, let alone avoid it. Ten straight forward hits, without taking a shot yourself, pushes you into the next screen where you have a quick change into a more horizontal aspect before moving back to screen one's targets on an even more technicolour background. What else can I say? Still having played the game and written about it, I can't quite make up my mind whether it's good ro bad. One thing is for sure. It's extremely irritating to play and lose so quickly.

K.M.

SOFTWARE SPOTLIGHT

See-Saw

★ ★ ★ ★ ★
Quicksilva
£7.95
CBM 64 + joystick

IT IS A GREAT TREAT TO review a game in which you do not chase anyone, are not chased by weird and wonderful characters or fire at space ships ad nauseum. This game is none of these but is so simple to be both addictive and challenging.

You are the Grand Master in the guise of a little blue blob. Your fellow brothers have been captured by the wicked lord and his henchmen and your task is to rescue them from his castle. The evil lord stands on the parapet of his castle and hurls bricks trying to dislodge you from your see-saw. However, you can use these bricks to your advantage by using a falling brick to propel a stationary one from your see-saw in the same manner as some acrobats at the circus (principle of moments and all that). You can aim these propelled rocks and, with luck, dislodge the lord's henchmen from the walls. Rid the battlements of sufficient henchmen and you can get yourself propelled over the castle walls without being grabbed. The lord and his henchmen are cowards and once you are in the castle they flee enabling you to rescue your friends.



On the next level, some of the bricks are heavier than others thereby making your task more difficult. Since I am yet to progress beyond level 2, I cannot comment on how things proceed from there.

Graphically this game is simple but brilliantly effective giving a cartoon like quality. The sound effects are certainly different, the moving henchmen sounding like someone eating a rubber band for lunch (if you see what I mean. If not, try it). I should add that this program is written by the Andromeda software people, and their pedigree speaks for itself.

This is a game of dexterity and strategy, which, if you are seeking that unusual experience, is the one to try.

M.E.W.

Boulder Dash

★ ★ ★ ★ ★
Statesoft
£8.95
CBM 64 + joystick

SNAP! ROCKFORD CRACKS INTO action, ready and waiting to be steered through the caves in search of jewels. He has to collect the indicated amount of jewels before the mysterious door to the next cave is revealed on the screen. Of course you might not see it at once because the game has a fairly large scrolling screen. The main danger for Rockford in each cave are the boulders which drop into the tunnels that he excavates in search of the jewels. Although they drop predictably enough, they still make a tasty strawberry jam out of Rockford. Apart from the boulders there are the growing amoebas to block, fireflies to avoid, enchanted walls to activate, butterflies to turn into jewels, to name but a few delights of this excellent arcade game. There are sixteen caves altogether to work your way through and five difficulty levels for each cave. And there is a bonus life too. Every five hundred points you score adds to the measly three original lives Rockford starts the game with. What's more, he needs them.

K.M.

Strontium Dog and the Death Gauntlet

★ ★ ★
Quicksilva
£7.95
CBM 64 + joystick

IF IT NEVER ACHIEVES ANYTHING ELSE, this game must have one of the longest titles in the market. From the instructions the Strontium Dog is really a mutant called Johnny Alpha (does the author read Harry Harrison?). Our hero is on the trail of a couple of renegade mutants called the Stix Brothers. The action takes place on the Planet of the Renegades where we find Johnny shot down in no man's land. Hence, our man of the moment must cross the planet shooting, avoiding a wide range of nasties.

From the nature of the task, you won't be surprised to hear that the game is of the scrolling type. You are shown the side view of the planet. This steadily scrolls from right to left giving the appearance of general movement to the right. Scattered along the way are animated nasties, plants and the occassional rock. Collision with these has a nasty effect on our hero's strength. Excessive depletion of his power results in his demise. Occasionally rather useful objects yield valuable points. You can vary your speed across the planet, but this rapidly depletes your strength. If trouble looms you can use electro-flares



to dazzle the opposition or time bombs to move back from trouble.

Graphically the game is very nicely done with plentiful use of raster interrupts. Similarly the use of sound is effective and suits the scenario perfectly. Along the bottom of the screen is an overall view of the planet showing your progress.

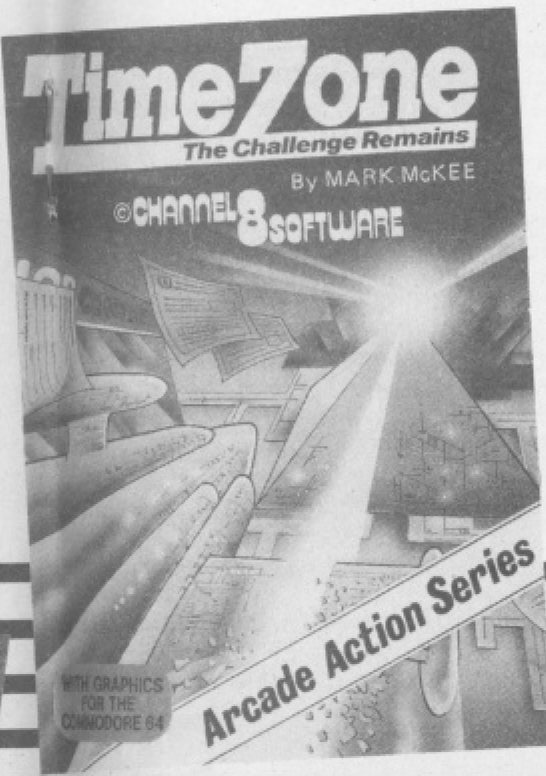
OK, how does it play? Well, at low levels it doesn't present a significant challenge and soon became rather boring. This is aggravated by the fact that completing one level simply puts you back at the start of the next level. In fact, after completing the first screen, I put the cassette to bed...permanently.

A.E.W.

Time Zone
★ ★ ★
Channel 8
£6.95
CBM 64

TIME ZONE IS ONE OF THOSE straightforward, honest-to-goodness zapping games with a playing zone which scrolls from right to left. You are the pilot of combat craft Alpha chasing the evil Terrilon out of the galaxy. The problem is recognising them when you see them because they are a race of shape changers. Their shape depends on the time zone in which they choose to travel. In the 20th century they attack as helicopters, UB40 cards, cruise missiles and tanks, in the medieval era as castles, arrows and winged horses. Classically they are temples and pyramids and in the pre-historic era pterodactyls, snakes and volcanoes. Presently they are drones and assorted space craft. The sole object of the game is to destroy them once and for all. But every time you clear all five sectors the difficulty level of the game will automatically increase. Can you win you may well ask? Well if you don't try who knows.

K.M.



Chiller
★ ★ ★ ★
Mastertronics
£1.99
CBM 64 + Joystick (Optional)

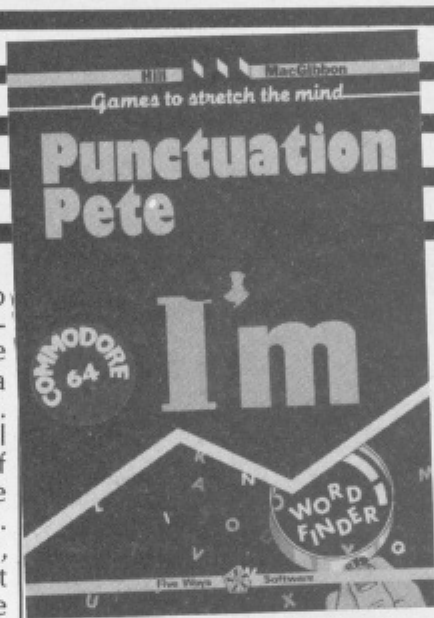
YOU'VE SEEN THE VIDEO, you've seen the making of the video, and you've heard the single. Now play the game! Yes, you too can be a werewolf with Mastertronics Chiller. I was, when I saw the package, very sceptical of this game. But now I've played it, it could get on to my top twenty favourites of all time.

I think the best way to describe it is as a high-resolution graphic realtime arcade adventure (if that's possible!). The basic idea is to rescue your girl friend from a house of the dead and get her back to the car. The problem is that you have to get through five screens of Zombies, collecting the magic crosses. Once you get to her you then have to get back (with her) to the car. Both you and the girl have to collect the crosses. The blue for you and the red for her.

Both characters are played with the same joystick by pressing the fire button and this automatically switches to the other character. The graphics are superb and the sound is very good.

S.L.F.P.





Punctuation Pete/Wordfinder
★ ★ ★ ★
Fiveways Software & Hill MacGibbon
£9.95
CBM 64

DID YOU KNOW THAT 'London' is a Celtic name meaning "the place of londinos" or that 'Pitlochry' is Pictish meaning stony farm or that 'Chester' comes from Ceaster meaning Roman fort? All these snippets came from the booklet accompanying 'Punctuation Pete'. This is one of a series of educational packages comprising a cassette and booklet. The booklet contains a potted history, cartoons and lots of follow-up activities, all relating to words and English.

The first part of the booklet covers story writing. Cartoons are used to illustrate the start of a story followed by a cartoon plus text for the next section. You are then invited to complete the story and pictures. There are other sections covering dialogue writing and a series of word games, crosswords, anagrams and code breaking.

The cassette contains two programs. The first, Punctuation Pete, comes with three difficulty levels and gives a passage for the child to correct. The keyboard is used to control Pete as he walks the lines of prose, stopping to allow the child to choose what to do. When the passage is finished, Pete will mark the work and if it is correct he will reward the child by jumping and waving. The routine used a large 'Primary school' type Font, which was both attractive and easy to read. The use of animation was smooth and skilful.

On the reverse of the tape was a Thesaurus which carried a dictionary of about 1000 words through which the child could discover the relationships between words and the concept of the simile. These words were used in hangman and anagram games.

Overall this is a good package which makes the English language fun (not an easy thing to do) and is suitable for use in both the classroom or at home.

M.E.W.

Felix In The Factory
★ ★
Micropower
£6.95
CBM 64 + Joystick (optional)

THIS IS ANOTHER VALIANT attempt at resurrecting the game of Apple Panic. it's a variation on a theme with an interesting new task but it's still basically a ladders game. The idea of this game is to keep the generator, at the bottom of the screen, running. This is accomplished by running up the ladders and getting the oil can. When you have returned to the generator and filled it up, the can disappears and you have to start again.

Whilst you are innocently maintaining this machinery, the Gremlins are trying to knock you off the platform which you hav to traverse to get the oil can. So by using the pitchfork, which is on one of the platforms, you try to knock them off the floors. Also, after a short time, a rat occasionally makes a dash across the platform. If you are lucky enough to have the bag of poison with you and you drop it in front of the rat, you gain extra points for killing it.

Meanwhile, at the bottom of the screen the generator is getting low on oil and, since you can only carry one thing at a time, it's a race to the can again. One thing to be careful of is the conveyor belt at the bottom on which you have to run because, also moving on the belt are numerous packages which you have to jump over. If you get knocked off or knocked over, you could lose a life. The sound isn't much to write home about but the graphics are passable.

S.L.F.P.

SOFTWARE SPOTLIGHT

Cuthbert Enters the Tombs of Doom
★ ★ ★ ★
Microdeal
£6.95
CBM 64 + Joystick

AFTER PRODUCING A HUGE volume of Dragon software, Microdeal are now into the CBM 64 market. This program continues the theme set on the Dragon by featuring Cuthbert, an overweight schoolboy with a taste for adventure. As clearly suggested by the title, this game is a graphical adventure set in a maze of interlinked rooms. (I believe that there are in excess of 200 of them!!)

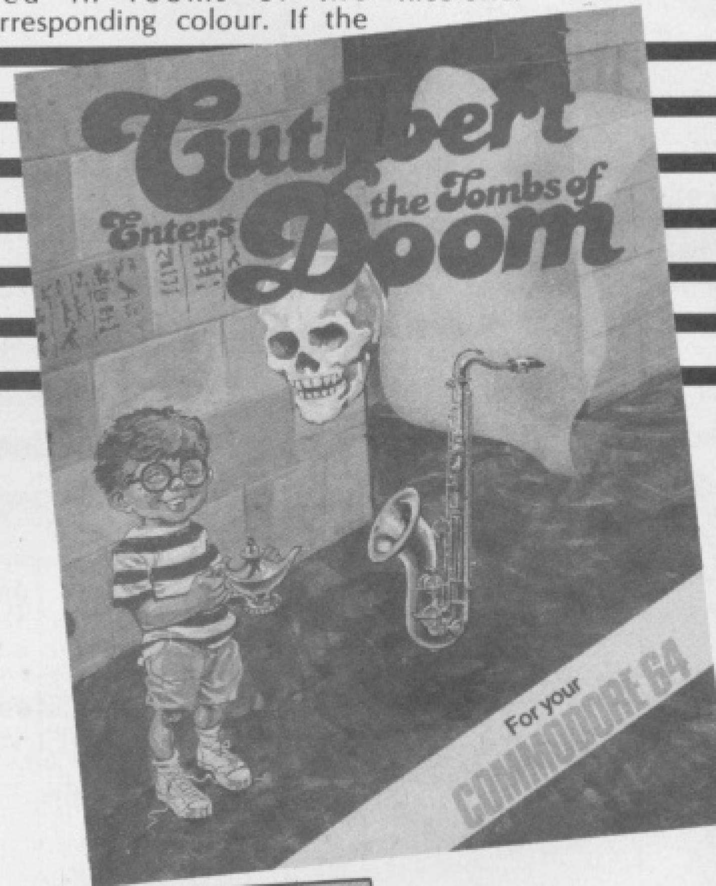
During his travels, Cuthbert must collect a number of things. First, keys are necessary to open doors barring his route. There are lanterns, rings, golden apples and urns which yield points. All of these items are found in niches in the walls.

Your travels are impeded in a number of ways. There are sundry nasties which appear in the rooms and will kill you given a chance. You are, of course, armed with a laser with which the enemy can be shot. The air between doors is limited and gradually decreases with time. You must therefore remember to unlock doors regularly before air runs out. There is one other method of protection. To aid your movement through some rooms, there are matter transporters. You have three coloured lamps which can be used in rooms of the corresponding colour. If the lamp is full, you can freeze the enemy once. You must then refill the lamp by scoring points. Filling a lamp also lights up letters in the word CATACOMB written at the top of the screen. When all of the letters are lit up, you get another life.

Overall this is a well designed game with tasteful graphics and a jolly rather tedious sound track. Certain of the rooms are designed so that cunning and tactics must be adopted to collect treasure or open doors. As with many games, the actual system for solution is fixed and can be resolved given time. You can, however, vary many of the game parameters to increase the difficulty of the game. My best is room 111 and that leaves me feeling exhausted. In all, a nice one.

A.E.W





Summer Games
★ ★ ★ ★ ★
Quicksilva
£19.99 (disc) £14.95 (cassette)
CBM 64 + Joystick

I HAVE JUST SPENT A VERY energetic afternoon swimming, diving, pole vaulting, gymnastics, running and skeet shooting and all without leaving the comfort of my computer corner. Quicksilva's new game jumps on the Olympic bandwagon and does it very well. The program starts with the opening ceremony. The flame is lit and doves are set loose to fly across the screen. You are then invited to enter your name and choose the country you represent. An approximation to your national anthem will be played if you win. (Up to 8 players can participate). You can then elect to either try a single event or partake in all. If you wish you can even practice any events you choose.

In the diving you have four dives and you can vary the dive (and tarrif) by deft manipulation of the joystick. Next is the pole-vault which needs a lot of dexterity and timing to clear the bar. The next events are the 400 metre relay and the 100 metre dash. Both races are against the clock as are the two swimming races.

The gymnastics are great fun especially as everytime I tried a vault the gymnast either landed on her face or rear end! The most difficult of all was the skeet shoot, probably because it was the most realistic. You have to shoot either single or double clays from several positions. The simulation was most accurate and even included the effects of shot scatter.

The graphics in this game are of a very high standard with detailed high resolution backdrops and excellently animated sprites. Overall this is an excellent program and is probably the best 'Olympic' program available.

M.E.W.

Swoop
★ ★ ★
Micro-Power
£6.95
CBM 64 + Joystick (Optional)

CAN THIS BE TRUE? I'M SURE it isn't — I think it is. Yes folks, it's Galaxians time again. Swoop is yet another imitation of that old game which people just wont let go. The difference between this one and others is that if a Galaxian, sorry, Space Vulture gets past you it lays an explosive egg in your path. Which means if you touch it you blow up or if you don't you could be cornered and get blown up anyway!

As you progress through the levels the Galaxians, cops, space vultures get more ferocious and it does require a lot of agility on the higher levels. Yet again, the music on this Micro-power offering is outstanding and a nice graphics touch has been added. The stars in the background move at different rates, giving an impression of depth and when your craft explodes the debris is scattered in a wide arc.

The game has different skill levels and a high score facility. There is also a pause button. So in all fairness it's not a bad copy of Swoop, sorry, Galaxians!

S.L.F.P.

THE 64 SOFTWARE CENTRE
1 Princeton Street, London WC1
01-430 0954

Business accunts

Sales Ledger (Anagram) d	75.00
Purchase Ledger (Anagram) d	75.00
Cashbook (Anagram) d	75.00
Sales Ledger (Ramtop) t	14.50
Purchase Ledger (Ramtop) t	14.50
Accounts package (Ramtop) d,t	115.00
Cashbook (Microsimplex) d	172.50
Cashbook (Gemini) d	64.95
Cashbook (Gemini) t	59.95
Final Accounts (Gemini) d	64.95
Final Accounts (Gemini) t	59.95
Cashbook & Final A/Cs (Gemini) d	99.95
Cashbook & Final A/Cs (Gemini) c	89.95
Inventory 64 (MMS) d	29.95
Stock Control (Gemini) d	24.95
Stock Control (Gemini) t	19.95
Stock Control (Anagram) d	75.00

Home applications

Budgeteer (Adamsoft) t	8.95
Checkbook Manager (Adamsoft) d	14.95
Home Accounts (Gemini) d	24.95
Home Accounts (Gemini) t	19.95
Home Office (Audiogenic) t	14.95
Home Accounts (Fieldmaster) d	29.95

Database systems

Magpie 64 (Audiogenic) d	75.00
Mailpro 64 (Pro-Line) d	79.35
Oracle 64 (B.I.) d	125.00
Practifile 64 (MMS) d	44.50
Simply File 64 (Simple) d	69.00
Superbase 64 (Precision) d	89.95
Figaro 64 (Saxon) d	86.25
Gemini Database d	24.95
Gemini Database c	19.95

Spreadsheets

Busicalc (Supersoft) d,t	17.95
Calcresult Adv. (Handic) d	99.00
Easy Calcresult (Handic) d	49.95
Multiplan 64 (Hesware) d	99.95
Practicalc 64 (MMS) d	44.50
Practicalc 64 (MMS) t	39.95

d = disk t = tape c = cartridge

ALL IN STOCK NOW — Callers welcome
(10am—6pm incl Saturdays)

Mail Order: Cheque or PO — add £3 for items over £30, others free
Credit cards (Access/Visa): phone or write. Immediate despatch

The 64 Software Centre, (Ref Y.C.)
1 Princeton Street, London WC1

Please supply the following items:

1 .... Qty..... @£ ....
2 .... Qty..... @£ ....
3 .... Qty..... @£ ....
4 .... Qty..... @£ ....

TOTAL £ ....
POSTAGE £ ....
TOTAL £ ....

Name .... Address ....
Visa/Access Card No ....
....
Date .... Signature ....

Programmers' Aids & Utilities

Supabasic (Interceptor) c	8.95
BC Basic (Kuma) c	57.50
Cadpack (Adamsoft) d	35.00
Chartpak 64 (Adamsoft) d	24.95
Forth 64 (Audiogenic) c	29.95
Forth 64 Advanced (Hesware) c	50.00
Graphics Designer (Adamsoft) d	19.95
Jetpack (DTL) d	39.95
Jetpack (DTL) c	14.95
Logo (CBM) d	57.50
Master 64 (Supersoft) d	71.30
Mikro Assembler (Supersoft) c	59.80
Hesmon Monitor (HesWare) c	34.95
Pilot (CBM) d	75.00
Printlink 64 (Supersoft) t	32.20
Quickchart 64 (Adamsoft) t	6.95
Screen Graphics (Adamsoft) d	14.95
Simons Basic (CBM) c	47.50
Sprite Aid (Adamsoft) t	6.95
Stat 64 (Handic) c	29.95
Superdisk Utility (Adamsoft) d	12.50
Synthy 64 (Adamsoft) d	14.95
Turtle Graphics II (HesWare) c	54.95
Ultrabasic 64 (Adamsoft) d	19.95
Ultysynth (Quicksilva) t	14.95
Victree 64 (Supersoft) c	56.35
64 Doctor (MMS) d	19.95
Doodle (Quicksilva) d	14.95
Songwriter (Scarborough) d	34.95
Master Composer (Access) d	44.95
Music Master (Supersoft) d	19.95

Word Processors

Heswriter (Hesware) c	39.95
Paperclip 64 (B.I.)	98.90
Simply Write 64 (Simple) d	46.00
Simply Write 64 (Simple) t	40.25
Vizawrite 64 (Viza) d	79.95
Vizaspell 64 (Viza) d	59.95

Educational & Languages

Essential Italian (Dell) d	16.95
Essential Italian (Dell) t	14.95
Besieged (Sulis) t	9.95
Wordpower (Sulis) t	9.95
Time Traveller (Sulis) t	9.95
Just a Mot (Sulis) t	9.95

various subjects: Callisto, Collins, Commodore etc.

War simulations

Battle for Normandy (SS) d	34.95
Knights of the Desert (SS) d	34.95
Eagles (SS) d	34.95
Tigers in the Snow (SS) d	34.95
Geopolitique (SS) d	39.95

Flight simulators

Flight Path 737 (Anirog) t	7.95
Interdicter Pilot (Supersoft) d	19.95
Interdicter Pilot (Supersoft) c	17.95
Flight Simulator (IFR) d,t	24.50
Flight Simulator II (Sublogic) d	47.50
Solo Flight (US Gold) d,t	14.95
Heathrow (HC) t	7.95

Business Simulations

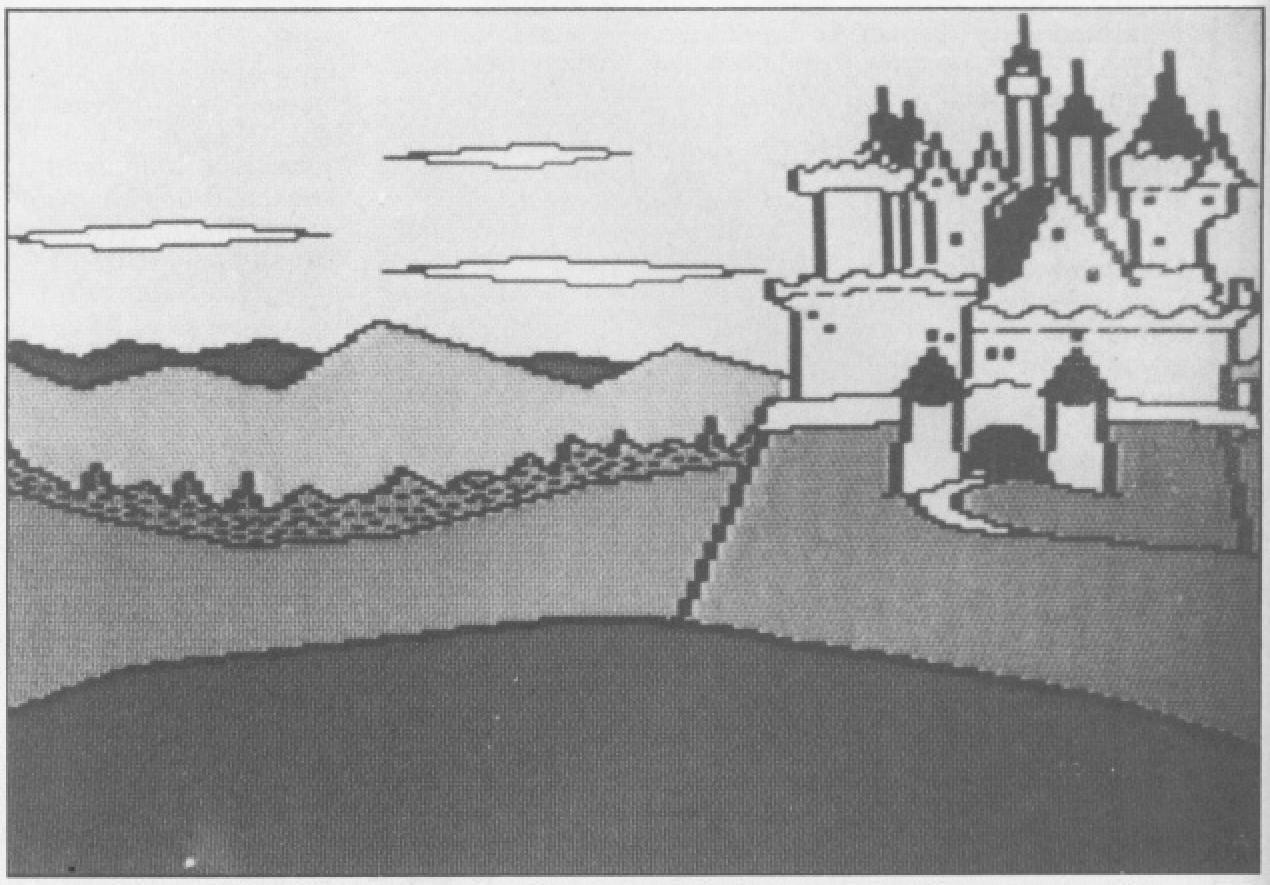
Commodities (Blue Chip) d	50.00
Real Estate (Blue Chip) d	50.00

Games

A large assortment in disk, cassette and cartridge.

Exploit the graphics capabilities of the Commodore 64 with the right tools. Simon Palmer and David Crisp review two graphic aids currently available.

DRAWN TOGETHER



KOALA PAD TOUCH TABLET
Koala Technologies Corporation
£79.95
Commodore 64

I KNOW KOALAS COME from Australia, but this one is definitely American, being an art package of a high calibre. It comes from the Koala Technologies Corporation, the software is written by Audio Light UK and distribution is in the care of Audiogenic.

"Koala Pad" Touch Tablet.

Attractively packaged in a bright red box, Koala's touch tablet comes complete with operating instructions for the software and touch tablet, plus a stylus for drawing on the touch-tablet.

Readers might find useful the following brief explanation of how a basic touch-tablet works. Imagine two sets of wires running north to south, and east to west. When the stylus is pressed onto the board it pushes down on to the wires and contact between the horizontal and vertical wires induces x and y coordinates in one form or another. These are then translated into signals which can be understood by the computer.

The software which is supplied is available on tape or disc and, as usual, it took a long time to load from tape. In this interval I took the opportunity to flick through the manual and immediately observed that it was written for the disc-based system, with just an insert for tape-users. However, since the Koala tablet is so easy to use, the manual is gloriously short and concise!

The Koala pad itself is made of a very strong plastic, and it plugs into Joystick Port 1. A long cable facilitates armchair operation and therefore greater comfort. It uses two registers to gain the stylus location information from the pad. For example, here is a simple little routine which changes the colours of the screen as the stylus is moved around the touch-pad:-

```
10 A=PEEK(54297):B=PEEK
(54298)
20 POKE53280,A:POKE
53281,B
30 GOTO10
```

It's crude but effective!

Another application could be the creation of a new musical instrument by using two different voice registers on the SID chip. However, I am digressing from the main thrust of this article — it is so easy to indulge in these fascinating by-plays!

Koala Painter

In conjunction with the Koala Pad, the Koala Painter opens up a new world for the user who wants to create high-level graphics with relative ease. With quick pen strokes you can switch from facility to facility, from paint brush to paint brush. This ensemble is by no means a direct competitor to the Pluto Graphics machine, but for around £80, it is good value for money.

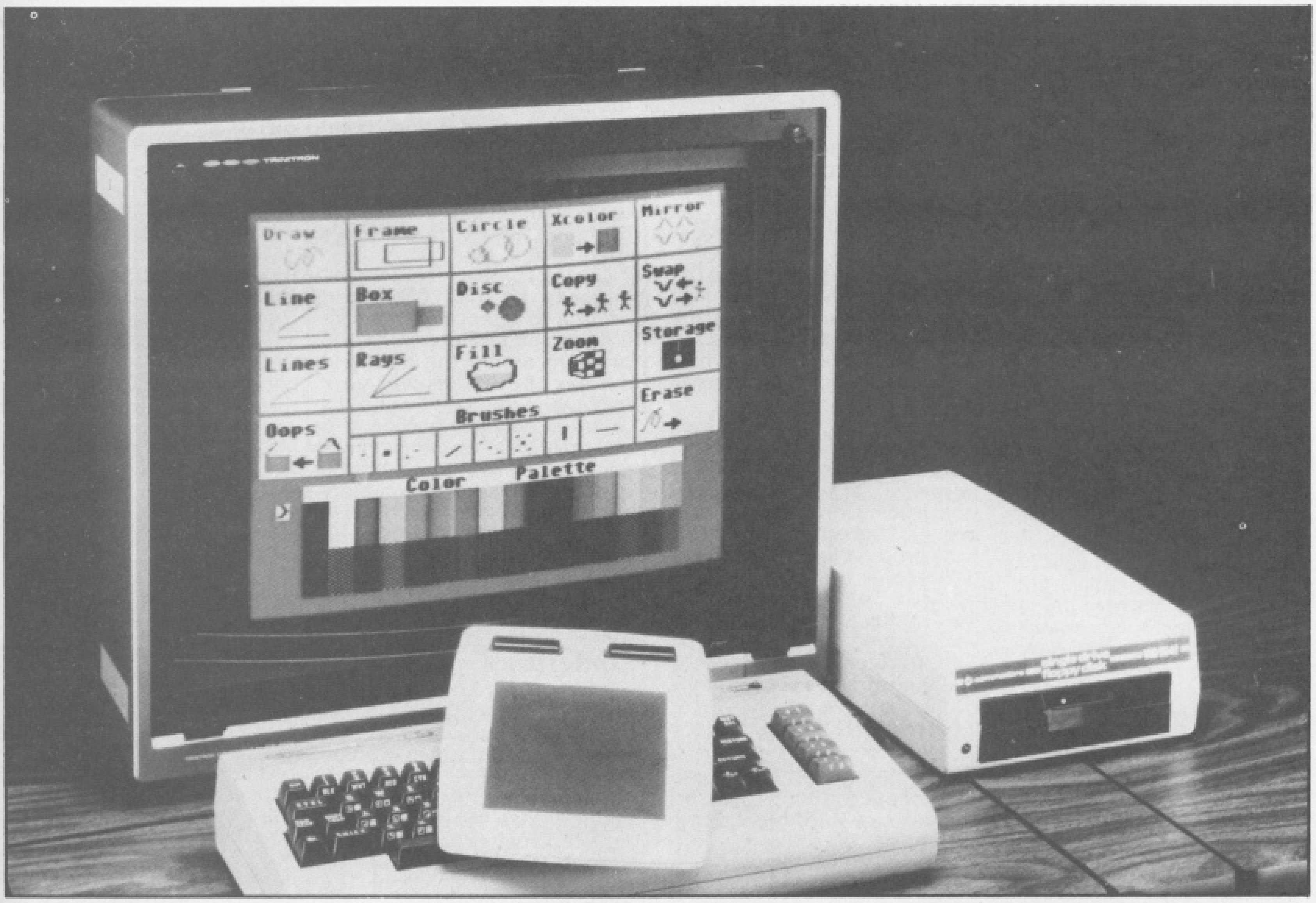
Once loaded, the machine displays the painter menu. It is divided into three sections: commands which flash when activated; brushes, under which relevant item a small block appears; colour palette on which 16 different colours are displayed, together with a further 16 patterned colours below them.

There are 17 different commands, of which 14 apply to graphics. The remaining 3 are facilities for erasing the total picture, saving or loading a picture, and erasing the last penstroke made.

When you enter the storage mode you can select either tape of disc for your medium. This facility also enables you to initialise a new disc without exitting the program. The "Oops" command is very useful indeed, because it is almost certain that at some time when you are using the painter, a mistake will be made. This command will remove the last penstrokes or "fills" made on the picture, enabling you to start it again. The Erase command simply removes the picture on the first screen, leaving it ready for another classic masterpiece.

The 14 graphics commands enable you to simply draw or construct a picture from the box or circle commands at will. A selection of the commands which I must mention are, "XCOLOR" (American spelling, not mine!) which will change a particular colour in the picture to another by nominating the new colour desired and then placing the cursor pointer over the target colour.

The next command is called "Mirror". This simply splits the screen into quarters and copies



whatever is drawn in one section into the others, line for line. However, if after completing a complex picture, you want to transfer a particular object which you have created to another space in the picture, the "Mirror" command will not suffice, as the object in question is already on the screen; this is where the "Copy" command comes into it's own.

Koala's "Copy" command reproduces the desired graphics and, since it is a square around the particular shape you're after, it also reproduces the graphics on the periphery of the command's action. Perhaps the user could benefit from this useful tip which I picked up whilst playing around with this section; this involves the use of another command called "Swap". Basically what this command does it to transfer the user from one screen to a second screen. I found that, by transferring the graphics I wished to copy to the second screen, and then by using the "Draw" command in the same colour as the background, I could eliminate the peripheral graphics not required, and then transfer it back to the original screen.

Finally, in the command section I decided to see how accurate the "Fill" command would be. The manual says that it can spill into other areas of the picture if the graphics are not completely enclosed. This is logical but attention is drawn to the fact that you can only use 3 colours and a background colour in a given space.

With this information in mind, I started to play with the "Fill". It is most efficient but colour-collision does occur when more than the stipulated number of colours are used. I feel that this is not so much due to the limitations of the software, but possibly the computer itself restricts the more ambitious inclinations of the exploring user! (Yes, even the CBM 64 cannot do everything!)

Brushing up

This package contains more brushes than Rolf Harris's paint box! Seriously though, 8 brushes are supplied in the Koala Painter from a single line to five lines, or even a very thick brush. So, in conjunction with some of the commands such as "Frame" you get some very pleasant designs to play with. Instead of getting a single box on the screen you get five or, if you wish, by using the "Draw" command, you can literally sign your signature on the screen. Mind you, it does not run fast enough to facilitate a normal writing speed but, at medium pace, it will give a fair representation. This is a good way of discovering how fast a package runs, and how accurate it can draw.

We now move on to an important part of the software — the "Colour Palette". It is most interesting to see how this section has been designed. There are 16 solid colours and 16 patterned colours from which to choose. If for example, you wished to have a patterned colour comprised of red and blue, you would move the pointer over the red solid square and press the button on the pad. The border of the screen turns red, and the patterned colours all now contain red as one of the two colours. You then simply move the pointer over the desired patterned colour which appears under the blue colour. You can now paint in the patterned colour, or more frequently it would be used to fill in a section of the screen. There is an arrow marker at the side of the Colour Palette which indicates the type of colour being used — either solid or patterned, thus you do not mistakenly use the wrong type and have to resort to the "Oops" command! In theory, since you have available 16 solid and 16 patterned colours, 256 combinations are achievable.

Pictures outside the program

It is most likely that, if you create a good picture, you will, like all artists want an exhibition for family and friends to admire your skill. You will also want to show it without all the hassle of loading the art program. Well, Koala have had the foresight to put a program in the manual which will enable you to load the picture from disc without using the Koala Painter. You may have already spotted that I said "disc" and not "tape" as well. I would have thought that since

most computer owners at present have only cassette and not disc, it would be more advantageous to include a tape version of the program. As I said earlier, the manual is written for disc users and not the cassette-tied populace. But even so, this does mean that if you are an adventure fanatic, you could design the graphics for your latest adventure, and by using the program in the manual, display them via the disc.

Now, throughout this article, I have not mentioned anything about example graphic screens. The manual and all the other bits of paper coming with the Koala Pad do not mention any extra screens so, obviously, there aren't any. Wrong — there is a picture contained in the tape directly after the program itself. This depicts a jungle scene and is quite good. So why did they not say something about it in the manual? I don't know — people are funny!

To conclude

Although this is not the only package available for graphics, considering all factors, plusses and minusses, at £80 or thereabouts it is jolly good value for money with a formidable array of commands and facilities summarized as follows:-

DRAW, FRAME, CIRCLE, XCOLOR, MIRROR, LINE, BOX, DISC, COPY, SWAP, LINES, RAYS, FILL, ZOOM, STORAGE, OOPS, ERASE.
8 Brushes
16 Colours + 16 Patterns (256 Combinations.)

S.L.F.P.

DESIGNER 64
Studio Software
£32.95
Commodore 64

QUOTING THE MANUAL, Designer 64 is a design and layout application program which incorporates a screen character layout editor and a program generator.

As all Commodore 64 owners know, this computer is capable of tremendous graphics with the right software. A lot of potential for good but simple graphics are built in via the keyboard but, if you know how, then greater things are possible.

I expected to be able to tap some of the 64's true potential from this piece of software, especially as it carries the much desired Commodore Approved Logo. It is possible that the product I received was pre-release as it came to me in a plastic bag with a label indicating what it was. The manual is substantial (A4) size, of considerable thickness and very clear and easy to read.

Aims and ambitions

With this program it was my intention to bang out some rather nice introduction pages to some of the programs I write. Also, I was going to lay out some day-to-day diary pages. I had success on the second job but not the first. Read on and I shall explain why.

My copy came with a second disc which had a few demonstration pictures, created with the software. The most interesting part of the package was that, after a design had been created, the program could translate what you had into a BASIC program, and that the BASIC listing could then be used in your own programs. To help achieve this, a merge routine could be used outside the program but, I must point out that it would have nothing to do with gosubs and gotos so its use is limited; however, used within the context of Designer 64 it is perfectly adequate.

Using these facilities, I designed my diary page and saved it to disc. I then wrote a little bit of BASIC around the program asking the date and the day of the week and, also, for how many days I wanted a printout. After inputting this information, it would print out pages of a diary which I often use. It is cheaper than a diary and, of course, I can have as many diaries as I choose. A useful application but hardly one to stretch a program to its limits.

Underused

The idea is to choose a shape. When you have chosen the shape you want, which must be a letter, number or graphics key, you then manipulate the shape. You can magnify, reduce, rotate and move it around the screen, fill the page with it and so on. The most important thing to note is that you are limited to the pre-defined Commodore key shapes. To my mind this was hardly adequate. It was not possible to draw true straight lines except by plotting a line of the relevant characters; an absence of commands such as circle, etc. meant that it was difficult to make anything look better than 'lumpy', for want of a better word. I can achieve almost the same results by clearing the screen, placing the characters I want to make the desired picture on the screen, putting a line number, question mark and a quote mark at the extreme left edge of the screen and pressing return. Repeat this down the screen and you then have your design as program lines: most important of all it costs nothing!

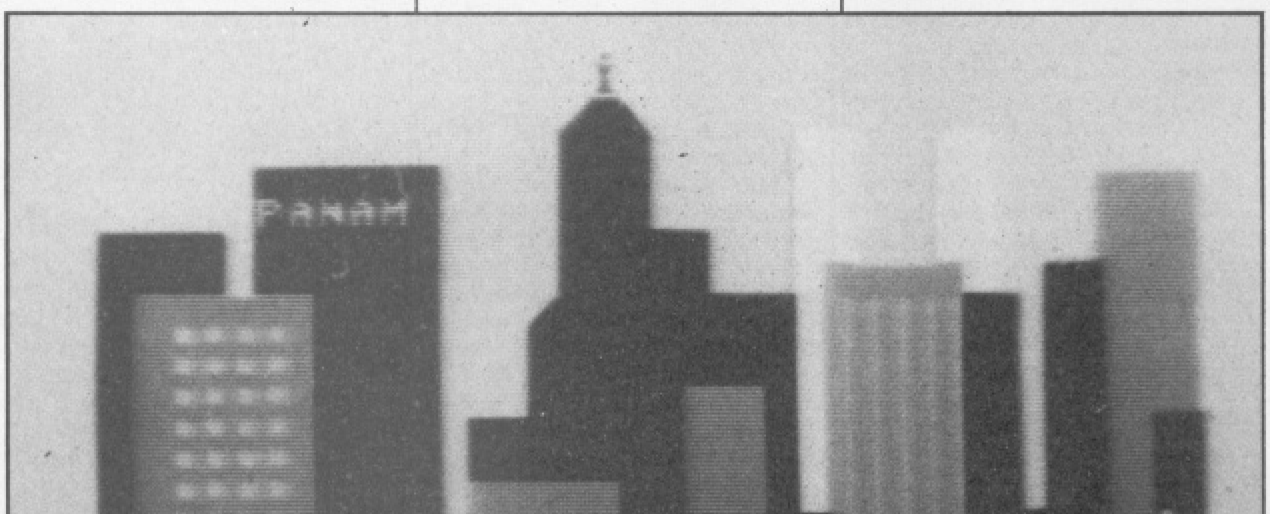
Admittedly a degree of animation is possible with Designer 64 but nothing that could not be done fairly simply through BASIC anyway. All this software does is take the graphic capabilities of the 64 at its lowest level, beef up the way you can move them around on screen, change colours and turn it into a finished subroutine. Deeper reading of the manual reveals that almost half of it is not concerned with actually using the graphics side of the machine but with disc use aspects and glossaries.

Without practice this was not an easy program to use and there was much to be remembered. Use of the keyboard and function keys was good though, and on screen prompts were logical. It was possible to get good screen dumps using a Commodore printer but there was absolutely no help given in the manual for the many people who use non-Commodore machines. If you want screen dumps from any of the more popular dot-matrix printers you would have to work out your own way of doing it.

More potential

Designer 64 is not an expensive program compared to some but it is certainly not in the pocket money bracket. To be honest, I don't think I would buy it unless I had a very specific use for its facilities. I must stress that it is very definitely a low-resolution drawing aid and therefore lacks a lot of potential. I get the feeling that it is a wasted opportunity and that for just a few pounds more it would be possible to get far more sophisticated design packages. Look closely at it before you buy as I feel that you may not be getting quite what you expect. To be fair though it does what it does very well and, if its design potential is adequate for what you want, then it is an excellent product and should make things a lot easier.

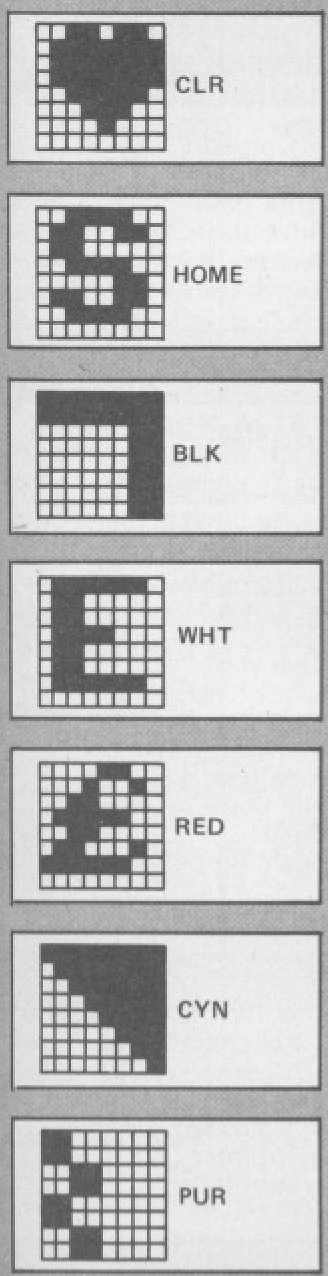
D.C.

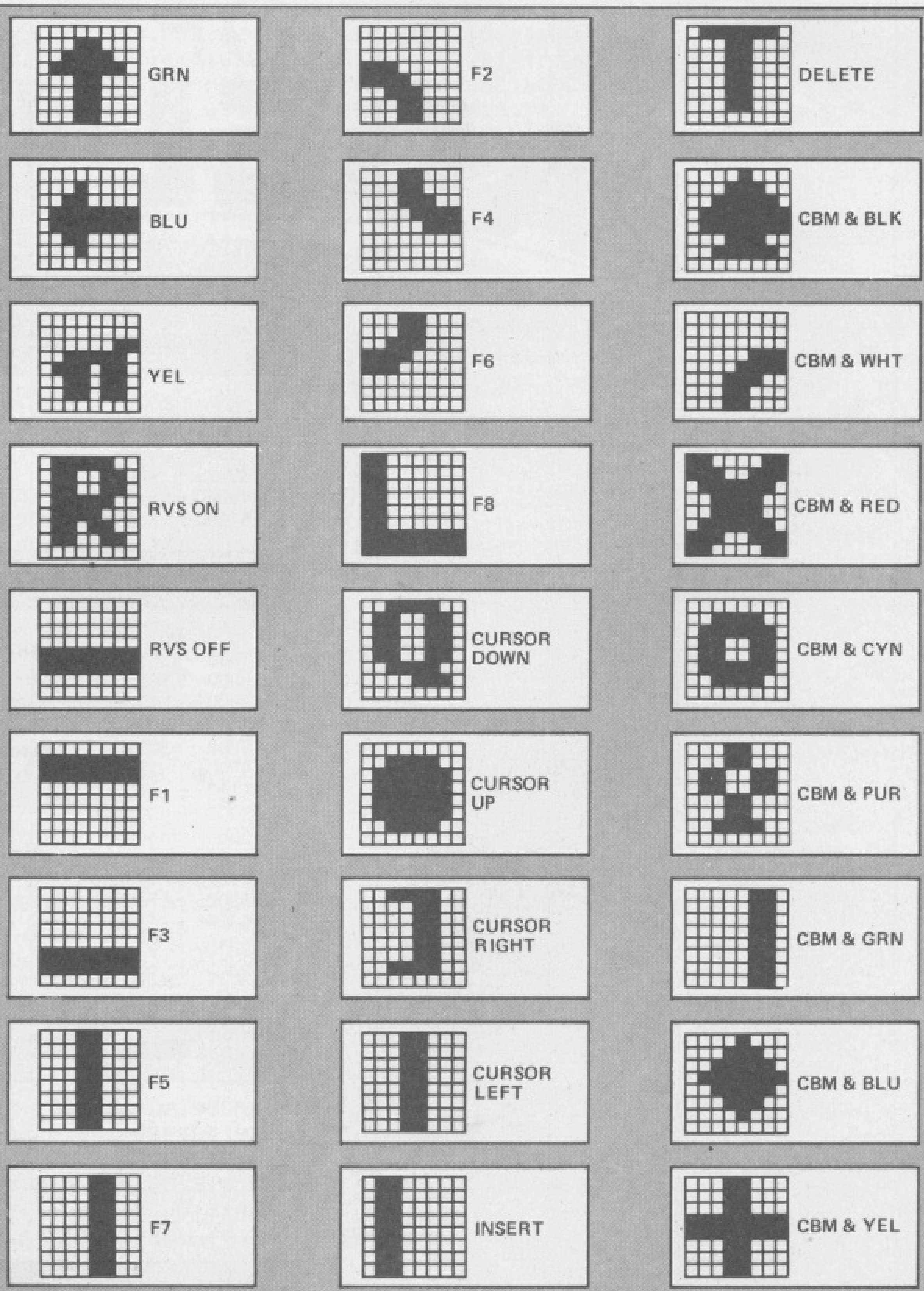


One of the most common queries received by 'Your Commodore' concerns the symbols found in the program listings. We hope the following information might help you.

64 SYMBOLS

THE 64 ALLOWS YOU TO specify control keys, etc. in print statements so that these functions can be executed within a program. For example, it is possible to position the cursor or clear the screen using the relative symbol with the quotation marks. The following list shows the symbols and the keys that generate them. (Don't forget the quotes).





THE FOUR CROWNS OF ADELIM

F.G. Tout invites you to recover the four crowns of Adelim and thus succeed where lesser mortals have failed.



IN THE PRE-COMMODORE days of old, Adelim ruled over the four kingdoms of Adelanthia. He held the four crowns and was a wise and powerful king. But he had an enemy — his name was Trewis. He planned and schemed until one day he slew Adelim and stole the four crowns. He took them back to his dark castle and hid them. Darkness has prevailed since that time, and all efforts to regain the crowns have failed.

One searcher alone, with courage and determination, might succeed where armies have failed...are you that searcher?

Instructions

In screen 1, I have used all characters as this makes it easier to itemise the collision detection but, of course, the character swimming tends to flash. However, from screen 2 onwards, it's sprites all the way — all 8 sprites, in machine code, plus fast, smooth screen scrolling on screen 4.

Screen 1

Swim around underwater, collecting all the treasure chests but don't hit anything else. When you have got all the chests, a key will appear. Pick up this key and the underwater gate will open, and allow you onto screen 2.

Insert joystick in port 2, move up/down, etc.

Screen 2

Jump the creatures and levels to get the crown.

Left = walk left : Right = walk right
Forward and fire button = jump

Screen 3

Climb the castle walls, avoiding the falling rotating blades.

Left = left : Right = right

Screen 4

Run the gauntlet, avoid the enemy soldiers and regain the crown.

Forward = up : Back = down

Screen 5

Step on and off floating platforms.

Left = left : Right = right

Part 1

20 — 499	Download U.D.G.s
600 — 960	Set pointer
1999 — 35099	Sprite data
39999 — 40020	Move sprites (sc. 1)
40040 — — —	Remmed★★★★★
61890 — 63500	Print water (sc. 1)

Part 2

10 — 100	Variables
1000 — 1030	Sc. 1
3000 — 5325	Move character (sc. 1)
6000 — 6100	Score
8000 — 8099	Sc. 2
9000 — 9099	Sc. 3
9200 — 9299	Sc. 4
9300 — 9399	Sc. 5
50000 — 50001	Clear msb
50005 — 50099	Routine for sc. 2
50100 — 50110	Jump (sc. 2)
61120 — 63005	Print water
63005 — 63050	Dec. lives
63200 — 63399	Title page
63400 — 63720	Intro
63750 — — —	Music (got crown)

Variables

V = 53248	(sp. variable)	Z	(charcter (sc. 1)
Y2	y pos sp. 1	SC	score
X2	x pos sp. 2	LI	lives
L	joystick	GG	pos sps 2-8
I&T	(variable)	Q	pointer (sp. 1)
CO	colour variable	S	sound reg
M	(variable)		

Program Listing Part 1

```
5 PRINT"♥":POKE53280,3:POKE53281,3:PRINT"                                      PLEASE  WAIT"
10 CC=0
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORG=0TO64*8:POKE14336+G,PEEK(53248+G):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
100 READA%:T=T+1:IFA%<>-1THENPOKE14336+G+T,A%:CC=CC+1:GOTO100
101 FORI=0TO60*8STEP8:A=14336+1+I:B=PEEK(A):POKEA,B AND1:NEXT
102 DATA0,0,0,0,0,0
105 DATA9,7,15,47,31,63,127,255,144,224,240,244,248,252,254,255
110 DATA16,36,24,88,61,126,126,255,247,0,222,0,123,0,222,0
115 DATA247,251,247,228,113,35,143,223,251,241,196,142,39,239,223,239
120 DATA239,223,239,39,142,196,241,251,24,102,129,255,129,129,129,255
125 DATA129,129,129,255,129,129,129,255,255,219,219,219,219,201,255,255
130 DATA7,31,127,251,219,219,201,255,224,248,254,223,219,219,201,255
135 DATA255,255,255,255,255,247,101,160,191,127,255,63,31,127,63,127
140 DATA5,166,239,255,255,255,255,255,254,252,254,248,252,255,254,253
145 DATA255,255,255,255,255,255,255,255,231,153,126,0,126,126,126,0
150 DATA126,126,126,0,126,126,126,0
160 DATA255,254,252,248,240,224,192,128,255,127,63,31,15,7,3,1
170 DATA165,195,129,90,60,90,145,145,153,90,60,255,255,60,90,153
175 DATA239,239,239,239,239,239,239,239,209,123,62,127,254,252,119,180
180 DATA254,170,85,106,254,198,238,254
185 DATA255,240,224,0,0,224,240,255,255,0,0,62,63,0,0,254
190 DATA240,0,6,63,63,0,0,255,240,0,6,63,48,0,6,255
200 DATA255,255,191,95,64,90,191,255
499 DATA-1
500 POKE53272,31
600 FORT=0TO2:FORX=0TO255
610 READA:CC=CC+1:IFA=-1THEN640
620 POKE36864+T*256+X,A:NEXTX
640 NEXTT
650 DATA169,146,141,21,3,169,0,141,20,3,96,-1
660 DATA169,234,141,21,3,169,49,141,20,3,96,-1
670 DATA172,0,147,192,17,240,57,206,1,147,173,1,147,201,0,208,44,238,0,147,169
680 DATA0,141,4,212,169,10,141,5,212,169,0,141,6,212,169,33,141,4,212,185,0,147
690 DATA185,0,148,141,1,212,185,0,149,141,0,212,185,0,150,141,1,147,76,49,234
700 DATA169,0,141,0,147,141,4,212,24,144,242,-1
710 POKE37632,0:POKE37633,0
720 FORT=0TO154:READHF,LF,D:CC=CC+1
730 POKE37888+T,HF:POKE38144+T,LF:POKE38400+T,D
740 NEXT:POKE37380,T+1
799 REM****MUSIC DATA*****
800 DATA19,63,8,22,227,8,21,154,8,17,37,8,19,63,8,22,227,8,21,154,8,17,37,8
805 DATA19,63,8,22,227,8,21,154,8,17,37,8,19,63,8,22,227,8,21,154,8,17,37,8
810 DATA25,177,8,25,177,8,22,227,8,21,154,8,22,227,6,25,177,6,28,214,6,32,94,8
815 DATA32,94,6,34,75,6,34,75,6,38,126,6,38,126,6,43,52,4,32,94,8,0,0,4,28,214,8
820 DATA0,0,8,25,177,8,0,0,8,17,37,8,17,37,8,19,63,8,22,227,8,21,154,8,17,37,8
830 DATA19,63,8,22,227,8,21,154,10,17,37,12,22,227,8,17,37,8,21,154,10,22,227,8
840 DATA25,177,8,25,177,8,34,75,8,34,75,6,38,126,8,22,227,8,21,154,8,22,227,8
850 DATA19,63,8,17,37,8,17,37,8,17,37,8,28,214,6,34,75,10,32,94,8,25,177,8
860 DATA28,214,8,34,75,8,32,94,8,25,177,8,22,227,8,22,227,8,21,154,8,17,37,8
870 DATA19,63,8,21,154,8,19,63,8,17,37,8,17,37,8,17,37,8,34,75,8,34,75,8
880 DATA43,52,8,43,52,8,38,126,8,34,75,8,32,94,8,28,214,8,28,214,8,32,94,8
890 DATA25,177,8,22,227,8,32,94,8,28,214,8,25,177,8,19,63,8,21,154,8,22,227,8
900 DATA22,227,8,28,214,8,28,214,8,25,177,8,25,177,8,22,227,8,28,214,8,28,214,8
910 DATA25,177,8,22,227,8,21,154,8,19,63,8,19,63,8,17,37,8,21,154,8,19,63,8
920 DATA17,37,8,17,37,8,32,94,8,32,94,8,28,214,8,25,177,8,28,214,8,32,94,8
930 DATA32,94,8,28,214,8,34,75,8,34,75,8,32,94,8,28,214,8,25,177,8,22,227,8
940 DATA25,177,8,28,214,8,0,0,8,28,214,8,25,177,8,0,0,8,25,177,8,22,227,8
950 DATA25,177,8,28,214,12,0,0,8,28,214,8,25,177,12,0,0,8,25,177,8,22,227,12
960 DATA0,0,8,22,227,8,21,154,12,0,0,8,21,154,8,19,63,8,19,63,12,17,37,100,0,0,8
```



```
1999 FORI=0TO767+256:READA:POKE205*64+I,A:CC=CC+1:NEXT
2000 DATA0,46,0,0,181,128,0,244,0,0,149,64,0,213,0,0,4,0,0,42,0
2010 DATA0,186,128,0,182,128,0,165,128,0,169,64,0,170,128,0,255,192,0,255,192
2015 DATA0,243,192,0,243,192,0,81,64,0,81,64,0,81,64,0,81,64,0,85,80,0
2020 DATA0,46,0,0,181,128,0,244,0,0,149,64,0,213,0,0,4,0,0,42,0
2025 DATA0,186,128,0,154,128,0,150,128,0,165,128,0,170,128,0,255,192,0,255,192
2030 DATA0,243,240,0,240,80,0,80,80,0,80,80,0,80,80,0,80,84,0,84,0,0
2035 DATA0,46,0,0,181,128,0,244,0,0,149,64,0,213,0,0,4,0,0,42,0
2040 DATA0,186,128,0,154,128,0,154,128,0,154,128,0,150,128,0,245,192,0,253,192
2045 DATA0,243,192,0,243,192,1,80,80,1,64,80,1,64,80,1,64,80,1,80,84,0
2050 DATA0,46,0,0,181,128,0,244,0,0,149,64,0,213,0,0,4,0,0,42,0
2055 DATA0,186,128,0,154,128,0,154,128,0,154,128,0,154,128,0,223,192,0,223,192
2060 DATA0,255,192,0,245,0,5,117,0,5,113,0,5,69,0,4,5,0,0,5,64,0
2070 DATA0,46,0,0,151,192,0,23,128,0,85,192,0,21,128,0,4,0,0,42,0
2072 DATA0,171,128,0,169,128,0,165,128,0,150,128,0,90
2075 DATA128,0,255,192,0,255,192,0,243,192,0,243,192,0
2080 DATA81,64,0,81,64,0,81,64,0,81,64,1,85,64,0
2085 DATA0,46,0,0,151,192,0,23,128,0,85,192,0,21,128,0,4,0,0,42,0
2090 DATA0,171,128,0,169,128,0,169,128,0,165,128,0,150,128,0,223,192,0,255,192
2095 DATA0,243,192,3,243,192,1,65,64,1,65,64,1,65,64,5,65,64,0,5,64,0
3000 DATA0,46,0,0,151,192,0,23,128,0,85,192,0,21,128,0,4,0,0,42,0
3005 DATA0,171,128,0,169,128,0,169,128,0,169,128,0,165,128,0,247,192,0,223,192
3010 DATA0,243,192,3,243,192,1,65,80,1,64,80,1,64,80,1,64,80,5,65,80,1
3015 DATA0,46,0,0,151,192,0,23,128,0,85,192,0,21,128,0,4,0,0,42,0
3020 DATA0,171,128,0,169,128,0,169,128,0,169,128,0,169,128,0,253,192,0,255,192
3025 DATA0,243,192,0,243,240,0,84,84,0,20,84,0,20,84,0,20,4,0,84,0,0
3030 DATA0,46,0,4,251,192,5,187,128,4,238,192,4,25,4,4,4,20,5,42,4
3035 DATA5,234,212,1,234,212,0,170,128,0,170,128,0,170,128,0,255,192,0,255,192
3040 DATA0,243,192,0,243,192,0,81,64,0,81,64,0,81,64,0,80,0,0,80,0,0
3045 DATA0,46,0,0,251,196,0,187,148,0,238,196,4,25,4,5,4,4,4,42,20
3050 DATA5,234,212,5,234,208,0,170,128,0,170,128,0,170,128,0,255,192,0,255,192
3060 DATA0,243,192,0,243,192,0,81,64,0,81,64,0,81,64,0,1,64,0,1,64,0
30001 DATA0,40,0,0,235,0,0,125,0,0,235,0,0,150,0,2,195,128,2,215,128
30004 DATA74,170,160,42,170,169,106
30005 DATA138,136,34,34,169,98,168
30006 DATA160,10,170,160,10,170,160
30007 DATA10,138,168,10,42,168,40
30008 DATA162,138,42,170,162,42,138
30009 DATA170,8,136,136,34,34,34,0
30011 DATA0,40,0,0,235,0,0
30012 DATA125,0,0,235,0,0,150
30013 DATA0,2,195,128,2,195,128
30014 DATA10,150,161,106,170,168,42
30015 DATA138,137,98,34,168,34,168
30016 DATA161,74,170,160,10,170,160
30017 DATA10,138,168,10,42,168,40
30018 DATA162,138,42,170,162,42,138
30019 DATA170,34,34,170,136,136,136,0
30020 DATA0,0,0,2,0,32,8,0,128,32,10,0,128,8,0,32,40,0,40,32,0,8,100,0,10,100
30032 DATA0,2,86,0,0,100,128,0,100,128,0,32,160,0,32,40,0,8,8,0,8,2,0,8,8,0,32
30034 DATA32,0,128,0,2,0,0,0,0,0,0
30036 DATA0,0,0,0,16,0,0,64,0,3,0,0,8,0,0,8,42,128,8,32,40,8,152,4,8,84,1,2
30038 DATA154,0,0,84,128,0,152,128,64,32,128,16,32,128,10,128,128,0,3,0,0
30040 DATA4,0,0,16,0,0,0,0,0,0,0,0,0,0
30042 DATA0,0,128,0,3,176,0,15,240,0,1,176,0,5,124,0,1,124,12,1,108,56,0
30044 DATA192,250,150,160,56,21,104,12,5,104,0,3,170,0,3,234,0,3,250,0,3,250,0
30046 DATA3,58,0,3,58,0,1,90,0,0,82,0,0,80,0,1,80,0
30048 DATA0,0,128,0,3,176,0,15,240,0,1,176,0,5,124,0,0,124,0,1,108,12,0,192,56
30050 DATA6,160,250,149,104,56,5,104,12,3,170,0,3,234,0,3,250,0,3,250,0,3,58,0
30052 DATA3,58,0,5,26,0,4,22,0,4,4,0,20,4,0
34000 FORI=0TO127:READA:POKE245*64+I,A:CC=CC+1:NEXT
34001 DATA0,0,0,0,0,0,0
34002 DATA0,0,0,0,0,0,0
34003 DATA0,0,0,0,0,0,0
34004 DATA0,0,0,0,0,0,170
34005 DATA170,168,153,153,152,170,170
34006 DATA168,15,3,192,53,205,112
34007 DATA48,204,48,208,116,28,170
34008 DATA170,168,166,102,104,170,170
34009 DATA168,0,0,0,0,0,0,0
35001 DATA0,0,0,0,0,0,0
35002 DATA0,0,0,0,0,0,0
35003 DATA0,0,0,0,0,0,0
35004 DATA128,192,128,226,162,192,170
35005 DATA234,128,186,171,128,222,237
35006 DATA192,187,123,128,136,200,128
35007 DATA170,170,128,119,55,64,252
35008 DATA15,192,112,3,64,192,0
35009 DATA192,64,0,64,0,0,0,0
39999 FORI=0TO48:READA:POKE49152+I,A:NEXT
40000 DATA 238,2,208,238,2,208,238,4,208,238,4,208,238,6,208,238
40010 DATA 6,208,238,6,208,206,8,208,206,8,208,206,10,208,206,10
40020 DATA 208,206,10,208,206,12,208,206,12,208,206,12,208,206,12,208,96
40040 FORI=0TO46:READA:POKE49650+I,A:CC=CC+1:NEXT:REM***SCROLL SCREEN*****
40050 DATA169,40,162,24,133,87,169,4,133,88,160,0,177,87,133,89,200,177,87,136
40060 DATA145,87,200,152,201,39,208,244,165,89,145,87,165,87,24,105,40,133
40065 DATA87,144,2,230,88,202,208,220,96
40070 FORI=0TO29:READA:POKE49500+I,A:CC=CC+1:NEXT:REM*****J/S*****
40075 DATA173,0,220,201,126,208,3,32,142,193,201,125,208,3,32,156
40080 DATA193,32,172,193,32,242,193,32,56,194,32,156,194,96
```

```
40085 FORI=0TO27:READA:POKE49550+I,A:CC=CC+1:NEXT:REM*****ME UP & DOWN*****
40090 DATA173,1,208,201,110,240,6,206,1,208,206,1,208,96,173,1
40095 DATA208,201,198,240,6,238,1,208,238,1,208,96
40100 FORI=0TO57:READA:POKE49720+I,A:CC=CC+1:NEXT:REM*****MOVE SOLDIERS*****
40110 DATA206,2,208,206,2,208,206,2,208,206,4,208,206,4,208,206,4,208,206,6,208
40120 DATA206,6,208,206,6,208,206,8,208,206,8,208,206,10,208,206,10,208
40130 DATA206,10,208,206,10,208,206,12,208,206,12,208,206,12,208,206,12,208,96
40150 FORI=0TO29:READA:POKE49580+I,A:CC=CC+1:NEXT:REM*****ME RIGHT*****
40160 DATA173,0,208,201,50,208,7,173,16,208,41,1,208,18,24,238,0,208
40170 DATA208,9,173,16,208,73,1,141,16,208,24,96
40200 FORI=0TO22:READA:POKE49820+I,A:CC=CC+1:NEXT:REM*****CHANGE SOLDIERS*****
40210 DATA162,1,189,248,7,24,105,1,201,221,144,2,169,219,157,248,7,232,224
40230 DATA7,208,236,96
40300 FORI=0TO50:READA:POKE50000+I,A:CC=CC+1:NEXT:REM*MOVE & CHANGE BLADES*****
40310 DATA162,4,189,1,208,24,105,4,157,1,208,201,245,144,6,173,4,220,157,0,208
40320 DATA232,232,224,16,208,231,96
40330 DATA162,2,189,248,7,24,105,1,201,219,144,2,169,217,157,248,7,232,224
40340 DATA8,208,236,96
61890 GOSUB62000:IFCC<>1969THENPRINT"DATA ERROR?":END
61900 FORI=0TO7:POKE14336+I,255:NEXT:POKE53281,3:PRINT"■J":POKE53281,3
61905 POKE53280,6
61910 SYS49900:FORI=1024TO1208:POKEI+14336,0:NEXT:GOTO63000
62000 FORI=0TO40:READA:POKE49900+I,A:CC=CC+1:NEXT
62001 RETURN
62010 DATA234,169,40,133,250,169,4,133,251,162,128,160,0,138,145,250,200,192
62020 DATA40,208,249,232,165,250,24,105,40,133,250,165,251,105,0,133,251
62030 DATA224,152,208,228,96,0
63000 U=15544:J=255
63010 FORU=15545TO15448STEP-1:POKEU,J:NEXT:J=60:FORU=15448TO15447STEP-1
63015 POKEU,60:NEXT:FORU=15447TO15446:POKEU,24:NEXT:PRINT"■J"
63500 POKE198,2:POKE632,13:LOAD
READY.
```



Program Listing Part 2

```
5 GOSUB63550
10 POKE54296,15:SYS36864
40 V=53248:POKEV+21,255:X2=46:Y2=205:POKEV+28,255:POKEV+37,10:POKEV+38,2
42 FORI=40TO46:POKEV+I,11:NEXT:DEFFNC(C)=(PEEK(V+31)AND1):POKEV+39,6:LI=5
44 L=56320
50 DEFFNA(A)=PEEK(X+M):DEFFNB(B)=PEEK(X+1+M):DEFFND(D)=(PEEK(V+30)AND1)
55 DEFFNE(E)=PEEK(V+1)
60 F$="■ ——————————————————————————— "
61 T$="          —        —         —         —"
65 G$="■■■—■■■":POKE53272,31
100 A$="■■■■■■■■■■■■"
101 GOTO63200
111 GOTO1000
1000 PRINT"J":GOSUB61120:PRINT"■■■":POKE53248+21,0
1001 FORI=1984TO2023:POKEI,69:POKEI+54272,13:NEXT:LE=1
1002 PRINT"■¯ — |":FORT=0TO2:PRINT"■■■|¯—¯":NEXT:PRINT"¯—■/L■—"
1005 FORT=0TO2:PRINT"—¯■\■|":NEXT:PRINT"■——————————":PRINT"■———————————■X"
1007 PRINT"■——————————":PRINT"■—■■|■■—■■■■■■■—■■—■■|■■¯";
1010 PRINT"■|—|¯|¯—¯|¯—¯—¯—|—¯|¯|¯—|¯—¯|¯—¯|—|¯"
1012 PRINTTAB(3)"■¯—|■■¯—|—■■■¯—|—¯■■■■■|■■■¯"
1015 PRINTTAB(12)"■¯|¯|■■¯|—|—■■■■¯|■■¯|—|—¯■■■■¯|¯■■—|■■■|¯—■■|"
1017 PRINTTAB(22)"■■■¯—|¯■■|—■■■¯—■■—¯■■■¯—|■■¯"
1020 PRINTTAB(34)"■■■—¯|¯—¯|¯—■■■■■—|¯—|¯■■■■■|—|¯■■■■—¯■■■—"
1025 PRINT"■■■■■"A$"■■■■■■■■"A$"■■■■■■■■■■■■"A$
1027 PRINT"■■■■×■■■■×■■■■■×■■■■■×■■■■■X"
1028 PRINT"■■■×■■■■■■×■■■■■■■×■■■■■■×■■■■■×■"
1029 PRINT"■■■■+■■■■■+■■■■■■■+■■■■■■+■■■■■■+"
1030 PRINT"■■■■■■■■■+■■■■■■■■+■■■■■■■■+■■■■■■+■■■■■■■■+";
3000 X=1524:CO=54272:M=0:Z=91:Z1=92:L=56320:QQ=81:ZZ=90:GOSUB6000
3010 POKEX+M,Z:POKEX+M+CO,3:POKEX+1+M,Z1:POKEX+1+M+CO,3:P=PEEK(L)
3011 IFM=180THENGOSUB63750:GOTO8000
3015 Z1=Z1+1:IFZ1>94THENZ1=92
3020 IFP=125THENGOSUB5020
3022 IFP=126THENGOSUB5120
3024 IFP=119THENGOSUB5220
3026 IFP=123THENGOSUB5320
3040 IFFNA(A)<81ORFNB(B)<81THEN63000
3041 IFFNA(A)>81ANDFNA(A)<90THEN63000
3042 IFFNB(B)>81ANDFNB(B)<90THEN63000
3050 IFFNA(A)=90ORFNB(B)=90THENSC=SC+1000:GOSUB6000
3051 IFFNA(A)=95THENPRINT"■■■■■■■■■■■■■■■■●"
4000 GOTO3010
5020 M=M+40:POKEX+M-40,81:POKEX+M+CO-40,3:POKEX+1+M-40,81:POKEX+1+M+CO-40,3
5025 RETURN
5120 IFM<20THENRETURN
```



```
5122 M=M-40:POKEX+M+40,81:POKEX+M+CO+40,3:POKEX+1+M+40,81:POKEX+1+M+CO+40,3
5125 RETURN
5220 M=M+1:POKEX+M-1,81:POKEX+M+CO-1,3
5225 RETURN
5320 M=M-1:POKEX+1+M+1,81:POKEX+1+M+CO+1,3
5325 RETURN
6000 PRINT"■SC ";SC;"      - LI";LI:D=D+1:IFD=10THENPOKE1933,95:POKE1933+CO,2
6010 RETURN
6100 PRINT"■":POKE53281,1:FORI=0TO10:NEXT:POKE53281,14:GOTO1000
8000 PRINT"■■■";:POKE53280,0:POKE53281,0:PRINT"■■■"T$:PRINT"■":LE=2
8005 PRINTF$:PRINTTAB(2)G$:PRINTF$:PRINTTAB(36)G$:PRINTF$
8010 FORI=1944TO2023:POKEI,81:POKEI+54272,3:NEXT
8015 PRINT"■■■":POKE2040,205:Q=205:POKEV+2,100:POKEV+3,145:POKE2041,215
8018 FORI=2041TO2046:POKEI,215:NEXT:V=53248:POKE2047,246
8020 POKEV+3,207:POKEV+4,225:POKEV+5,207:POKEV+6,40:POKEV+7,143:POKEV+8,150
8021 POKEV+9,143:POKEV+16,128:POKEV+14,60:POKEV+10,83:POKEV+11,79:POKEV+12,254
8022 POKEV+13,79:POKEV+30,0:POKEV+31,0:POKEV+21,255:X2=30:Y2=206
8025 POKEV+46,7:POKEV+15,49
8080 GOSUB50000:GOSUB50005:PRINT"■SC  ";SC
8099 GOTO8080
9000 POKE53280,11:POKE53281,0:POKE53272,31:PRINT"■";:LE=3:L=56320
9002 PRINT"■ .    .     .    . ■■|- ■   ..    .■-   ■.  ";
9003 PRINT"■    . ■- ■   . ■  -■■■■|■|- ■  .■-■|■■|■  .  .";
9004 PRINT"■ .- |■■-■|■■- ■ ■■-■■-■■-■■-■■|■| ■ |■■-■■-■|■-■- .  ";
9005 PRINT". -|--|--|-■|-■■■|-|---||-■|■■--|---||-■|   ";
9006 PRINT" ■■■-|-|--■■■■■■■--|--■■■■■■■--|--■■|. ";
9007 PRINT"■■■■■-|--|■■■■■■■■■■|■■■■■■■■■■|-||-■■■| ";
9010 FORT=0TO12:PRINT"■--|-|---|-|-|-|--|--|--|----|--|-";:NEXT
9015 FORT=0TO2:PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";:NEXT
9020 PRINT"■--------------------------------------"
9024 PRINT"■■■ ■■■■,"
9025 PRINTTAB(15)"■■ ■■■■■■■■,■■■■■■■■ ■■■■■■,■■■■■■■ ■■■■■,"
9030 PRINTTAB(18)"■■■\\\■■■■\\\■■■■\\\■■■■\\\■■■■ \_";
9040 Z$="■■■■■■■■■■■■■■■■■■■___":X$="■\_■■■\_■■■\_■■■\_■■■||"
9050 PRINTTAB(10)Z$:PRINTTAB(25)X$:PRINT"■■■":PRINTTAB(6)Z$:PRINT"■■■"
9052 PRINTTAB(2)X$:PRINTTAB(30)Z$
9070 FORI=1944TO2023:POKEI,81:POKEI+54272,3:NEXT:Q=213:POKE2040,Q
9073 POKEV+2,160:POKEV+3,50:POKE2041,246
9074 POKEV+29,252:FORI=40TO46:POKEV+I,1:NEXT:POKE50007,8:X2=60
9075 GG=255:POKE56335,1:FORI=5TO15STEP2:GG=GG-31:POKEV+I,GG:POKEV+I-1,0:NEXT
9076 POKEV+21,255:POKEV+30,0:X2=60:Y2=210:POKEV+40,6:POKE2041,246
9077 GOSUB50000:SYS50000:SYS50028
9079 Q=Q+1:IFQ>214THENQ=213
9080 POKE2040,Q:P=PEEK(L):IFP=119THENX2=X2+6
9082 Y2=Y2-.8:IFP=123THENX2=X2-6
9083 IFPEEK(V+1)<70THENPOKEV+21,0:POKEV+29,0:SC=SC+2000:GOSUB63750:GOTO9200
9085 IFFND(D)=1THEN63000
9087 PRINT"■■■";SC
9099 GOTO9077
9200 POKE53280,0:POKE53281,0:PRINT"■■";:LE=4:PRINT"■■SC■■";SC
9202 PRINT"■      .    O . .        . .  O    "
9204 PRINT"■   .  ■|              .        .  ";
9205 PRINT"■    .■■■|  .■|   . -     .  ■|  .    . ■|   ";
9210 PRINT" ■|■■■■■|■■■|   ■■|■|-■|■■■|  . - .■■■|■|";
9212 PRINT"■■■■■■■■■■■■■■|■■■■■■■■■■■■■■|■|■■|■■■■■■■■";
9215 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";:RR=0
9217 FORT=0TO2:PRINT"■--|--\_\\|__|--|--|--|-|-\\|--|-|--|-|";:NEXT
9220 FORT=0TO10:PRINT"                                        ";:NEXT
9227 FORT=0TO2:PRINT"■--|--|-| \\--|_--|--|--\_\--|-|_--|-|-";:NEXT
9230 V=53248:POKEV+21,255:POKEV,20:POKEV+1,198:POKE2040,205:Q=205
9235 POKEV+2,150:POKEV+3,180:POKEV+4,70:POKEV+5,120:POKEV+6,250:POKEV+7,130
9236 POKEV+8,60:POKEV+9,135:POKEV+10,116:POKEV+11,150:POKEV+12,70:POKEV+13,160
9237 POKEV+14,200:POKEV+15,198:POKEV+30,0:FORI=2041TO2047:POKEI,219:NEXT
9250 POKEV+16,128:POKEV+14,30:POKEV+15,150:POKE2047,246:POKEV+46,5
9255 Q=Q+1:IFQ>208THENQ=205
9257 POKE2040,Q
9260 SYS49500:IFPEEK(V)<40ANDFND(D)=1THENSC=SC+3000:GOSUB63750:GOTO9300
9265 IFFND(D)=1THEN63000
9299 GOTO9255
9300 PRINT"■":POKE53280,0:POKE53281,0
9304 FORI=2041TO2046:POKEI,245:NEXT
9305 V=53248:POKEV+2,60:POKEV+3,70:POKEV+4,60:POKEV+5,190:POKEV+6,60
9307 POKEV+7,160:POKEV+29,126
9310 POKEV+8,60:POKEV+9,130:POKEV+10,60:POKEV+11,100:POKEV+12,60:POKEV+13,210
9315 POKEV+16,128:POKEV+14,30:POKEV+15,180:POKEV,30:POKEV+1,52:POKEV+21,255
9320 L=56320:X2=30:Y2=53:Q=205:POKE2040,Q:POKEV+30,0:POKEV+31,0:POKEV+40,8
9322 PRINT"■■■■■--------■■■■■■■■■■■■                  ----------"
9325 PRINT"■■■■■■■■■■■■■■■■■■■■■■■-------------------------------------------"
9330 POKEV+14,45:POKEV+15,150:POKE2047,246:POKEV+46,7
9350 GOSUB50000
9352 P=PEEK(L):IFP=119THENQ=Q+1:X2=X2+4:IFQ>208THENQ=205
9354 IFP=123THENQ=Q+1:X2=X2-4:IFQ>212THENQ=209
9356 IFPEEK(V)>60ANDFND(D)<>1THENY2=Y2+4:L=0
9357 IFFND(D)=1THENL=56320
9358 SYS49152:POKE2040,Q:SYS49152:GOSUB50000:PRINT"■   ";SC
9359 IFY2>210THEN63000
9360 IFX2>290ANDFND(D)=1THENSC=SC+4000:GOSUB63750:GOTO1000
```

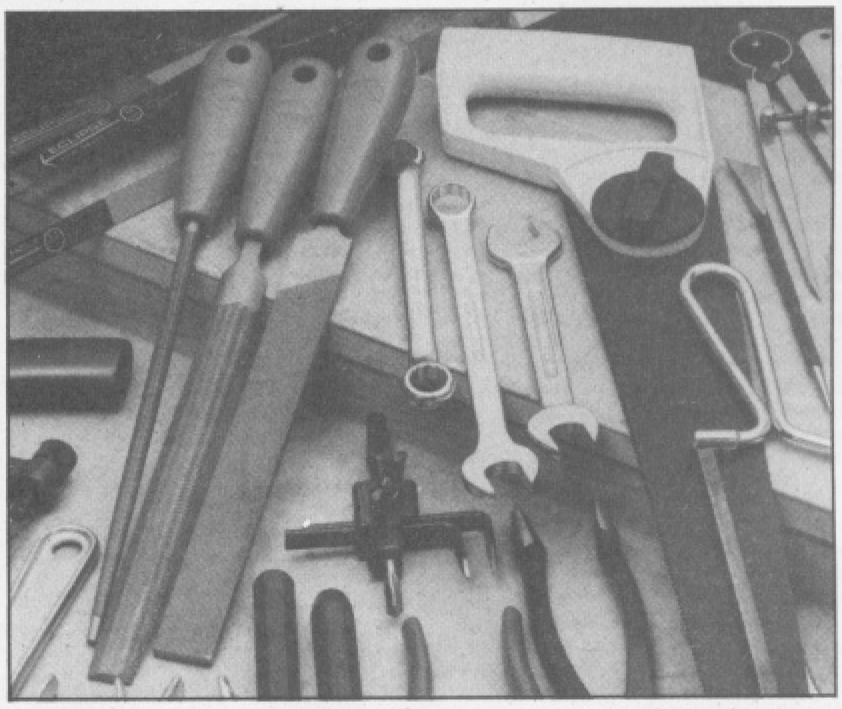
```
9399 GOTO9350
50000 POKEV,X2+256*(X2>255):POKEV+16,(PEEK(V+16)AND254)OR-(X2>255):POKEV+1,Y2
50001 RETURN
50005 REM
50006 POKE2040,Q:SYS49152
50010 P=PEEK(L):IFP=119THENX2=X2+6:Q=Q+1:IFQ>208THENQ=205
50020 IFP=123THENX2=X2-6:Q=Q+1:IFQ>212THENQ=209
50025 IFFND(D)=1THEN63000
50030 IFFNC(C)<1THENL=0:Y2=Y2+2.2
50040 IFP=110THENGOSUB50100
50045 IFFNC(C)=1THENL=56320
50050 IFPEEK(V+1)<60ANDPEEK(V)>255THEN9000
50099 RETURN
50100 Q=213:POKE2040,Q:L=0:POKEV+1,Y2:J=J+1:Y2=Y2-2.5:IFJ>12THENJ=0:RETURN
50101 F=F+1:IFF>1THENF=0:SYS49152
50102 IFJ>11ANDFNC(C)=1THENSC=SC+250
50104 IFX2>305ANDFND(D)=1THENSC=SC+1000:GOSUB63750:GOTO9000
50105 IFFND(D)=1THENEND
50110 GOTO50100
61120 POKE53281,3:PRINT"■▟":POKE53281,14:POKE53280,2:POKE53272,31
61130 SYS49900:RETURN
63000 V=53248:PRINT"▟":POKEV+21,0:POKE53281,1:POKE53280,1:FORT=0TO15:NEXT
63005 FORI=1TO15STEP2:POKEV+I,0:POKEV+I-1,0:NEXT:LI=LI-1:IFLI<1THEN63200
63010 ONLEGOTO1000,8000,9000,9200
63050 GOTO63010
63200 POKE53280,8:POKE53281,0:PRINT"▟":POKE53248+21,0:POKE53272,31:V=53248:SC=0
63210 PRINT"■■■■■■■               THE ■F■O■U■R ■CROWNS ■OF"
63250 PRINT"■■■■■■■ ●┌┬┬┬┐ ●┌┬┬┐   ●┌┬┬┬┬ ●┐      ●┐ ●|  ●|  ";
63255 PRINT" /●\\●┐ /●\\●| /●\\\\ /┐      /┐ /●|●●┐  ";
63260 PRINT" /┐  /┐ /┐  /┐ /┐     /┐      /┐ /●\\●┐  ";
63265 PRINT" /┐  /┐ /┐  /┐ /●┬    /┐      /┐ /┐  /┐  ";
63270 PRINT" /●┬●┐ /┐  /┐ /●\\    /┐      /┐ /┐  /┐  ";
63275 PRINT" /●\\●┐ /┐  /┐ /┐     /┐      /┐ /┐  /┐  ";
63280 PRINT" /┐  /┐ /●┬●| /●┬┬┬ /●┬┬┬ /┐ /┐  /┐  ";
63285 PRINT" ⁄\  ⁄\ ⁄\\┘  ⁄\\\\\ ⁄●\\\\\ ⁄┐ ⁄┐  ⁄┐ O"
63286 PRINT"■■  BY FRANK TOUT   PRESS SPACE TO PLAY    "
63287 PRINT"■──────────────────────────────────────";
63288 PRINT"■──────────────────────────────────────";
63290 FORI=2040TO2043:POKEI,246:NEXT:E=1:FORI=39TO42:POKEV+I,E:E=E+1:NEXT
63295 POKEV,85:POKEV+1,100:POKEV+2,140:POKEV+3,100:POKEV+4,200:POKEV+5,100
63300 POKEV+6,255:POKEV+7,100:POKEV+21,15
63350 GETU$:IFU$=""THEN63350
63360 IFU$=" "THENPOKEV+21,0:LI=5:GOTO111
63399 GOTO63350
63400 PRINT"▟":POKE53272,31
63401 FORTT=0TO20:POKE53281,1:FORT=0TO50:NEXT:POKE53281,0:FORT=0TO50:NEXTT,TT
63410 FORT=0TO255:POKE53280,T:NEXT
63550 PRINT"▟":POKE53280,9:POKE53281,0:POKE53272,22
63600 PRINT"▟      |HE FOUR CROWNS OF ●─┐L.\.         ";
63605 PRINT"■     ------------------------------"
63610 PRINT"■▙      .............AND IN THE DAYS OF LIGHT";
63615 PRINT"■●─┐L.\  RULED  OVER THE FOUR KINGDOMS OF";
63620 PRINT"▙●DELANTHIA.  |E HELD THE FOUR CROWNS AND";
63630 PRINT"■WAS A WISE AND POWERFUL KING.           ";
63635 PRINT"▙       |UT ●─┐L.\ HAD AN ENEMY ,DARK AND ";
63640 PRINT"■MYSTERIOUS, |REWIS WAS HIS NAME.        ";
63645 PRINT"▙      |E PLANNED AND SCHEMED,AND THEN ONE";
63650 PRINT"■NIGHT ,SLEW ●─┐L.\ AND STOLE THE FOUR   ";
63655 PRINT"▙CROWNS .  |E TOOK THEM TO HIS DARK CASTLE";
63660 PRINT"■AND HID THEM."
63665 PRINT"▙       ─ARKNESS HAS PREVAILED SINCE THAT ";
63670 PRINT"■TIME,AND ALL EFFORTS TO RETRIEVE THE    ";
63675 PRINT"▙FOUR CROWNS HAVE FAILED."
63680 PRINT"■      ┌NE SEARCHER ALONE WITH COURAGE AND";
63685 PRINT"▙DETERMINATION MIGHT SUCCEED WHERE ARMYS ";
63690 PRINT"■HAVE FAILED.  ●RE YOU THAT SEARCHER?... "
63695 PRINT"■           ┐L─●● ●┐●── |┌ ─┐ ⁄─/        "
63700 GETA$:IFA$=""THEN63700
63710 IFA$=" "THENRETURN
63720 GOTO63700
63750 SYS36864+256:FORN=54272TO54296:POKEN,0:NEXT:POKE54296,15:S=54272
63775 POKES+5,177:POKES+6,154:RESTORE
63780 READHF,LF,D:IFD=0THENSYS36864:RETURN
63785 POKES+1,HF:POKES,LF:POKES+4,17:FORT=1TOD*25:NEXT:POKES+4,16
63790 GOTO63780
63800 DATA19,63,8,22,227,8,28,214,8,25,177,8,19,63,8,22,227,8,28,214,8,25,177,8
63805 DATA28,214,8,34,75,8,34,75,8,38,126,8,34,75,8,34,75,8,34,75,8,38,126,8
63810 DATA38,126,8,43,52,8,38,126,8,34,75,8,28,214,8,43,52,8,38,126,8,34,75,8
63815 DATA28,214,8,22,227,8,25,177,8,28,214,8,25,177,8,22,227,8,17,37,8,22,227,8
63820 DATA17,37,8,17,37,8,22,227,8,17,37,8,17,37,10,22,227,10,22,227,8,25,177,8
63825 DATA25,177,8,28,214,8,34,75,8,34,75,6,34,75,4,38,126,6,38,126,8,38,126,10
63830 DATA38,126,8,43,52,8,43,52,6,38,126,8,34,75,8,28,214,8,0,0,4,28,214,8
63835 DATA22,227,8,22,227,6,28,214,8,25,177,8,22,227,8,21,154,8,22,227,8,28,214
63840 DATA8,22,227,8,21,154,8,19,63,10,21,154,8,22,227,8,21,154,8,22,227,8
63845 DATA25,177,8,22,227,8,22,227,10,22,227,20
63990 DATA0,0,0
READY.
```



In the fourth part of this series, Grahame Davies shows you how to write your business programs to disc.

DOING IT YOURSELF

IF YOU HAVE BEEN FOLLOWING this series, then by now you should have nicely finished programs to enter data and present it in a neat fashion. To store this data we will use a disc drive. Of course, you may wish to use a cassette recorder to store data on, but this is not to be recommended for several reasons: firstly, and most obviously, Commodore cassettes are slow, secondly you can practically only store data in a sequential format and so to read one particular record, you will always have to start at the first record and scan through the file, thirdly you cannot update one record without re-writing the complete file and so the only practical way to use a cassette is to read the complete file into the computer at the start of the program and to write it all back at the end of the program; doing this will severely limit the amount of data you can store.



Sequential disc files

Sequential files on disc have the same qualities as sequential files on tape but are faster. There are good reasons for using them though — for instance you may require a transaction list of some sort and would never need to look at a single record so a sequential file would be suitable. To make this clearer, say you were writing a program to take care of your record collection. You would have a file of record names together with details about those records and you would want to access an individual record so a sequential file would not be suitable. However, if you wanted to know where all your records are (who you have lent them to and so on) then every time you tell your program you have lent a record out, it would make an entry on the transaction file. Every record returned would also have an entry on this file and each new entry would be added to the end of the file. When you come to list this file you will simply read it from the start and print out every entry. Reading a sequential file from start to end is faster than doing the same with relative files (see below) and sequential files are easier to maintain.

Most of you reading this article will be using a 1541 disc drive and if you are then you must be aware that it has a fault which can easily be (and must be) avoided. When overwriting a directory entry using the '@' feature, the file may become corrupt. This applies to program files and sequential files. To avoid this, always use the scratch, rename and save facilities:

```
scratch filename + ".bak"
rename filename to filename
+ ".bak"
save filename
```

You will notice that these are *not* BASIC commands but just general guidelines to the saving of files. Another advantage of doing it this way is that you end up with a back-up version of the file being saved.

Relative files

Relative files can look difficult and clumsy to use at first but are in fact a very good way to store data allowing plenty of flexibility. If you are using a machine with BASIC 4.0 instructions then the following conversions will be of no interest although they will still work. For the rest of us with BASIC 2.0 as on the VIC 20 or the CBM 64 all we need to do is tackle the awkward commands once and create them as subroutines. Having done this they will become simple to use. The end result of using a relative file is that every time you update a record, it is always immediately written back to the disc which means that starting the program and ending the program does not involve lengthy waits while data is being read in or written back. This also means that we can store unlimited amounts of data (you just have to buy another or a bigger disc drive if you run out of space).

To explain the principal of a relative file, take a plain sheet of paper and down the left hand side from top to bottom write down the integers 1 to 20. Each of these integers represents a record. The blank space to the right of the record represents the data that is stored in that record. If we select a record, say number 3, then we can immediately go to record number 3 and start writing data in it. We carry on writing to it until we reach the far edge of the paper when we run out of room. The amount of data we have now written, represents the record length. We could write in every record of the sheet (the file) until we reached the bottom and then we would take another sheet of paper and so on. If we wanted to write so much data against one record that it would not fit on that line, then we would have to throw that piece of paper away and start with a new and wider piece. In practice of course you would probably just squeeze an extra word or two in but when using disc drives you obviously cannot.

A relative file takes the same format as your piece of paper: at the start you decide how wide it is to be (the record length) and what name to call it. These two things cannot be changed later and so it is important that you get them right first time. If you try to write past the end of the file (off the bottom of the paper) then a relative file will automatically make itself bigger — this is called expanding. When we create a relative file, we will pre-expand it. The reason for this is that expanding takes extra time which you will not want when your program is running. Another point is that if you try to write past the end of the file, the disc drive will first of all return an error (50 — record not present) which is really just a warning message. When you

print to the file it will expand itself and the next time you access that record the error message will not be returned. This means that your error checking code will have to allow for this and this makes it messier.

Accessing a record

When accessing a record we use the equivalent of the RECORD command which is in the form RECORD file number, record number, byte position. The byte position is the position along the record to start from. If this is not sent, or is zero or one, then we will look at the record starting from the first character. If it is five then we will start character number five in that record. Of course, BASIC 2.0 does not have a RECORD command so there is a subroutine to replace it but if you think of it in the same terms as above, it will be easier to use.

Expanding a file

The following listing incorporates a subroutine to expand a file with the filename given in f$ (less than 17 characters long), the record length in r% (up to and including 254 characters) and the number of records to pre-expand the file to in s% (remember that you can always make it longer later on). To expand the file we will RECORD to position s% and print a chr$(255) there. The reason for this is that when a relative file is expanded, all of the empty records automatically get created with chr$(255) and you can use this fact to look for empty records within a file.

Note that the listing is split into three subroutines — to create a file, call line 2000, the routine at 2100 is a RECORD routine and there is a disc status checker at 2900. I suggest you make this a separate program which prompts you for the file name and so on. If you want to use the character position facility then I suggest you pass the subroutines another variable called bp% and change line 2140 to end with chr$(bp%) instead of chr$(1). Note this is for use in your main line code only and is NOT suitable for creating a file. If you alter line 2180 to read gosub2900:return then you can include the subroutines at 2100 and 2900 in your application program.

At this point I will just point out a few other things to be aware of with relative files: firstly you can only have one open to a 1541 at any one time because it takes three buffers and this drive only has five buffers, secondly always remember to close all your files when the program has ended even if it breaks in with a syntax error because the drive has a buffer which will only get written back when you force it to use the CLOSE command.

```
1998 rem
1999 rem
2000 open15,8,15:rem open command channel
2020 print"creating new file: "f$
2040 open3,8,2,f$+",l,"+chr$(r%):gosub2900
2060 dr=s%:gosub2100:rem record command
2080 print#3,chr$(255):gosub2900
2090 close3:close15:return
2098 rem
2099 rem
2100 rem record command
2120 a=int((dr+1)/256)*256:b=dr+1-a
2140 print#15,"p"chr$(2)chr$(b)chr$(a)chr$(1)
2160 rem send position = p, secondary address = 2
2170 rem hi and lo byte of record, character position 1
2180 return
2198 rem
2199 rem
2900 rem check disc error channel
2920 input#15,a,b$,c,d:ifa=0then2980
2940 print a,b$,c,d :stop
2980 return
ready.
```

Listing 1

Three subroutines

We now need three more subroutines to have a set of working utilities. The first will open an existing relative file whose name is given if f$. The second will read from record position dr into rc$ and the third will write a record in rc$ to record position dr.

```
2200 rem open an existing relative file
2220 open 15,8,15:rem open command channel
2240 open3,8,2,f$:gosub2900:rem open relative file
2280 return
```

```
2300 rem read in from file 3 secondary add = 2
2320 gosub2100:rem position to correct record
2340 inputL3,k$,rc$:gosub2900
2360 rem input and check disc status
2380 return
```

```
2400 rem write to file 3 secondary add = 2
2420 gosub2100:rem position to correct record
2440 printL3,k$chr$(13)rc$;:gosub2900
2460rem print to disc
2480 return
```

Note that line 2340 inputs two strings from disc. When using the INPUT command from disc, you will find that it behaves just the same as INPUT from the screen — so you can only input up to 80 characters (inclusive) and input will truncate on colons and carriage returns.

At this point you should practice using these routines. Note that when you use the PRINT command, the computer will always output a carriage return at the end of the statement unless their is a semi-colon ';' as in line 2440. The reason for suppressing this extra carriage return is that if you have a record length of 30 characters and you send a record that is 30 characters long, when you print this to the disc, the extra carriage return will have nowhere to be printed. The same thing occurs when you try to write a record to the disc that is longer than the record length of the file. This will cause an error 51, overflow in record and the extra data will be lost (and will *not* overwrite the next record). record).

Record keeping system

We can now put these routines to good use by creating a fast access record keeping system. The method to be used is called hashing and you will find that it is well suited to relative files. The principal is straight forward. When you want to call up an item from disc, you have to tell the computer something you know about that item and this information is called the key. A key can take any form you like — a part number, a name, letters and numbers mixed and so on. The only restricting factor is that to call up an item you must type its key exactly, as just one difference in the key will cause the program to look at the wrong part of the file. From this key, we will form a "random" number and we will use this number to calculate where to look in the file. The best thing to use is the Commodore's random number generator for this purpose but you must remember that this "random" number has to be calculated to the same result every time you wish to call up that record.

The next routine creates a random number from a key passed to it in k$. If you call this

routine with the same key it should always return the same number. You could of course write your own algorithm in here and for some applications this will be very important. This particular one scans along the key and takes the ASCII of every character and multiplies it by it's position in the key and then adds it into a variable 'a'. The reason that we must multiply by the position is so that a key "ABC" does not give the same result as "CBA" and so on. Having done this we seed the random number generator with the negative value and then take the random of that number and convert that into an integer between one and the file size which is in s%. The result is returned in dr.

```
2500 rem hash routine – given key in k$
2510 rem calculates hash position into dr
2520 a=0:fori=1tolen(k$)
2540 a=asc(mid$(k$,i)★i:next
2560 b=rnd(–a):a=rnd(b)
2580 dr=int(s%★a)+1:return
```

We now need one more subroutine to make use of this hashing. This will return a variable 'er' as to whether we find a record or not. Another variable returned is 'dc' and we will use this for deleting and creating purposes; I will explain this later on. Remember that when we create a relative file, all of the records have got chr$(255) in them and so to find a record we pass this routine the key and it creates the hash number and then scans the file until it finds the key or a chr$(255).

```
2600 rem find key
2605 rem er=0 found ok   dr=position
2610 rem er=1 not found   dr=position
2620 ky$=k$:gosub2500:rem preserve key & get hash
2640 dc=0:rem position of chr$(254)  (deletes).
2660 gosub2300:ifk$=ky$thener=0:return:rem found
2680 ifk$=chr$(255)thener=1:return:rem not found
2700 rem chr$(255) = empty record
2720 ifk$=chr$(254)anddc=0thendc=dr
2740 dr=dr+1:ifdr>s%thendr=1
2760 rem hash files are circular – see notes
2780 go to 2660
```

Line 2740 needs a little explaining. For our purposes we will have a file with one hundred records. If a key hashes to record one hundred and that is already occupied, then the next position to look at is record number one. From this you can see that there is no actual end to the file which is why it is called a circular file.

Using the routines

The following program gives an example of how to use these routines. Type them in and try them and I will then explain how they work and the importance of the delete flags.

```
readv.
5 f$="test":s%=100:gosub2200
10 print:print"create, amend, delete or end. c,a,d or e"
20 inputa$:ifa$="e"thenclose3:close15:end
40 a=0:ifa$="c"thena=1
50 ifa$="a"thena=2
60 ifa$="d"thena=3
70 ifa=0thenprint"Error":goto10
80 input"Key";k$:ky$=k$
90 onagosub100,200,300:goto10
100 gosub2600:ifer=1then140
120 print"Key already exists":return
140 input"Create - data";rc$
160 k$=ky$:ifdcthendr=dc
180 gosub2400:print"created at"dr:return
200 gosub2600:ifer=0then240
220 print"Key not found":return
240 print"amended at"dr
260 print"Amend Data    "rc$
270 input"<CRSR UP><CRSR RIGHT*11>";rc$
280 gosub2400:return
298 rem
299 rem
300 gosub2600:ifer=0then340
320 print"Key not found":return
340 k$=chr$(254):rc$="deleted"
360 gosub2400:return
999 close3:close15:end
```

Listing 2

RUN the program, type 'c' for create and type in the key as "abd" and any data that you want to be the record. Note that the program expects a relative file called "test" to be set up with a minimum of one hundred records and a record length of about fifty bytes. When this key "abd" is created, it is created at position 42. This will not be true if you typed the key in upper case or if you mixed lower and upper case. If you want to avoid problems with upper and lower case then change line 2540 to this:

```
2540    a=(asc(mid$(k$,i))and
127)★i:next
```

You should be able to type 'a' for amend followed by a key of "abd" and amend this record. You cannot of course amend the key as it will move on the file. Now create a key of "aaa" and you will find that it is created at 55. So far so good but now to test the problem of hashing to an occupied position. Create a key of "dcr". This key hashes to 42 which is occupied by "abd" and so it is created at position 43.

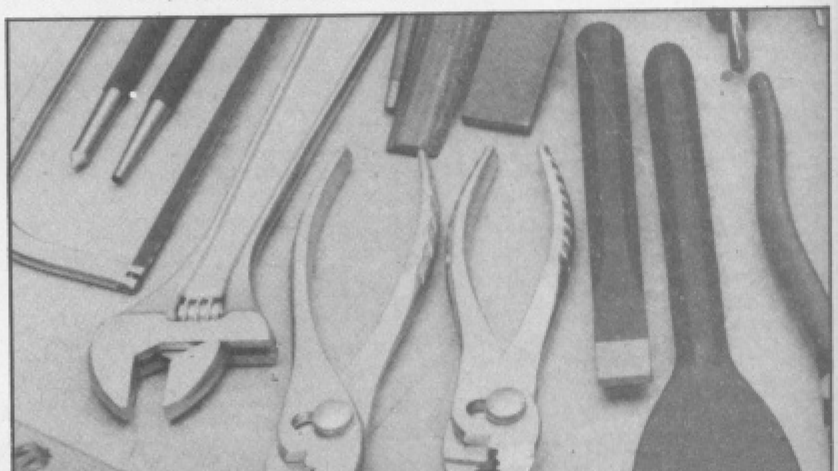
Deleting records

We now have a method of creating and amending records using keys. Unfortunately, deleting is slightly more complicated. The obvious thing to do when deleting is to place a chr$(255) in the deleted record position. If we did this and deleted the key "abd", we would not be able to call up key "dcr" because this key would hash to 42 which is now a chr$(255) and the routine would say that that is an empty record and stop searching. If you don't believe this then change line 340 to read:

```
340  k$=chr$(255):rc$=
"deleted"
```

Now delete "abd" and try to amend "dcr". Having done this change line 340 back to what it should be. To overcome this we have a special flag on the disc. I have chosen to use chr$(254) as you are very unlikely to have this as your key — if you did use it as a valid key, you would cause havoc with your data! Using the above example, deleting "abd" will cause a chr$(254) to be written to the disc at position 42. The routine for scanning for a key at 2600 will treat this as any other key and *not* assume end of file. The only special handling in there is the variable 'dc'. This is set to zero on entry and if the scan finds a chr$(254) it is set to that disc position. Note that on one scan, it only sets 'dc' to the first chr$(254) found as there may be others. This flag is then used in the create part in line 160. The create routine logically says: check the key does not already exist : if not then enter the data : now create the record at dr which is the position of the chr$(255) found by the scan routine *unless* while we were scanning we found a chr$(254) — if so then create it there at that position (dc).

Using the above methods you should have a reasonable fast access system. As more items are put on, the access time will slow but should remain acceptable until the file becomes about seventy per cent full. This will depend on how good your hashing algorithm is and a certain amount of luck as to whether the keys used create a good spread across the file.



THE KEY TO PROFESSIONAL GAMES DESIGN

ON THE SPECTRUM 48K

THE GAMES LANGUAGE OF THE 80'S!

Now, without any knowledge of machine code, you can write fast, smooth, professional, totally original games and market them **without paying royalties.**

Even if you have already mastered machine code, we believe that the time and problems saved by writing in **White Lightning's** FORTH– based high level language could revolutionise commercial games writing for years to come.

IDEAL IDEAL is an Interrupt Driven Extendible Animation sub-Language. Once you have mastered IDEAL's easy to learn set of over 80 commands and just a little FORTH, you will be ready to produce arcade-quality games even if you don't know machine code. Up to 255 Sprites, each with its own user-defined dimensions can be moved around the screen (or memory), scrolled, spun, reflected enlarged or inverted with amazing speed and smoothness. Operations are possible between screen windows, Sprites and Sprite windows. Sprites can even stretch across several screens, so those difficult scrolling landscapes that form the basis of so many games are easy to achieve. Sinclair's own sound and graphics commands such as CIRCLE, DRAW and BEEP are fully supported, and there are some unique collision detection facilities.

MULTI-TASKING Because **White Lightning** uses interrupts, you can effectively run two programs at once. This means of course, that games like Space Invaders and Defender can be written without complex timing calculations. So while one program smoothly scrolls the landscape, the second animates the other characters. This is undoubtedly one of White Lightning's most powerful features.

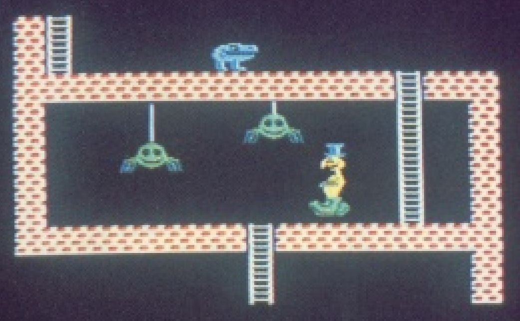
WHITE LIGHTNING

MARKETING AND PORTABILITY Although **White Lightning** uses an integer FORTH as its host language, programs can be written in a combination of BASIC, FORTH, IDEAL and machine language.

What is more, programs written in FORTH/IDEAL will be highly portable between the Spectrum and implementations under development for other popular micros.

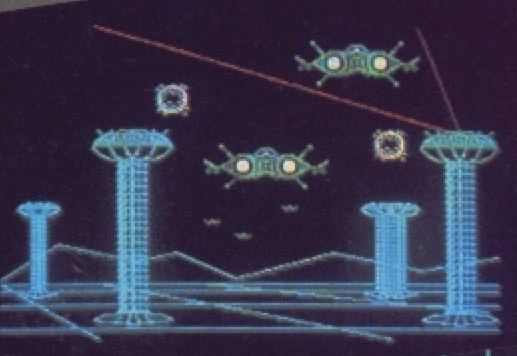
When it comes to marketing your completed games, there's no problem either. In fact Oasis themselves will offer to market outstanding software.

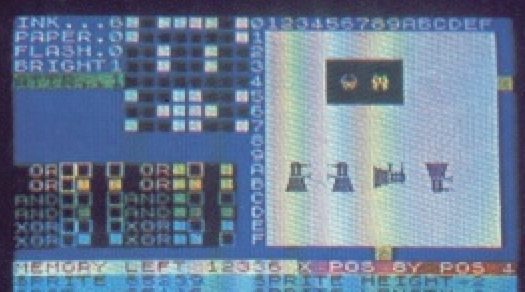
SPRITE DESIGN White Lightning, comes complete with a separate 20K program for developing the Sprites used in the main system. Not only can you use this to design your own Sprites from scratch, it also comes complete with 168 pre-defined characters covering games like **Asteroids, Pac-Man, Assault Course, Defender, Space Invaders, City Bomber, Lunar Lander, Frogger, Centipede, Donkey Kong** and many, many, more. These characters are ready to use or can be enhanced. And Sprites can be saved to tape between editing sessions before being finally loaded into the main program.

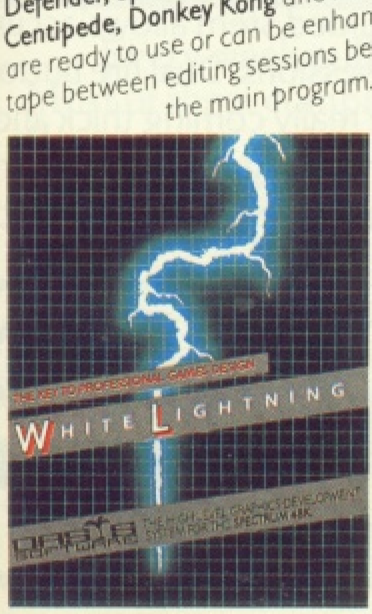




- Produces real machine code programs which run independently of White Lightning.
- A multi-tasking animation language AND a Sprite Development program together in one system-pack.
- Supplied with a FREE 16-minute demo, and a 130-page, easy-to-follow manual.







The High Level Graphics Development System for the SPECTRUM 48K . . . . .

OASIS SOFTWARE

AND COMING SOON! the power of White Lightning on the COMMODORE 64!

If your local dealer doesn't stock White Lightning, just send off this coupon for our rapid-despatch service.

Please send me ___ White Lightning System Packs at £14.95 each. I enclose my cheque/P.O. for £______

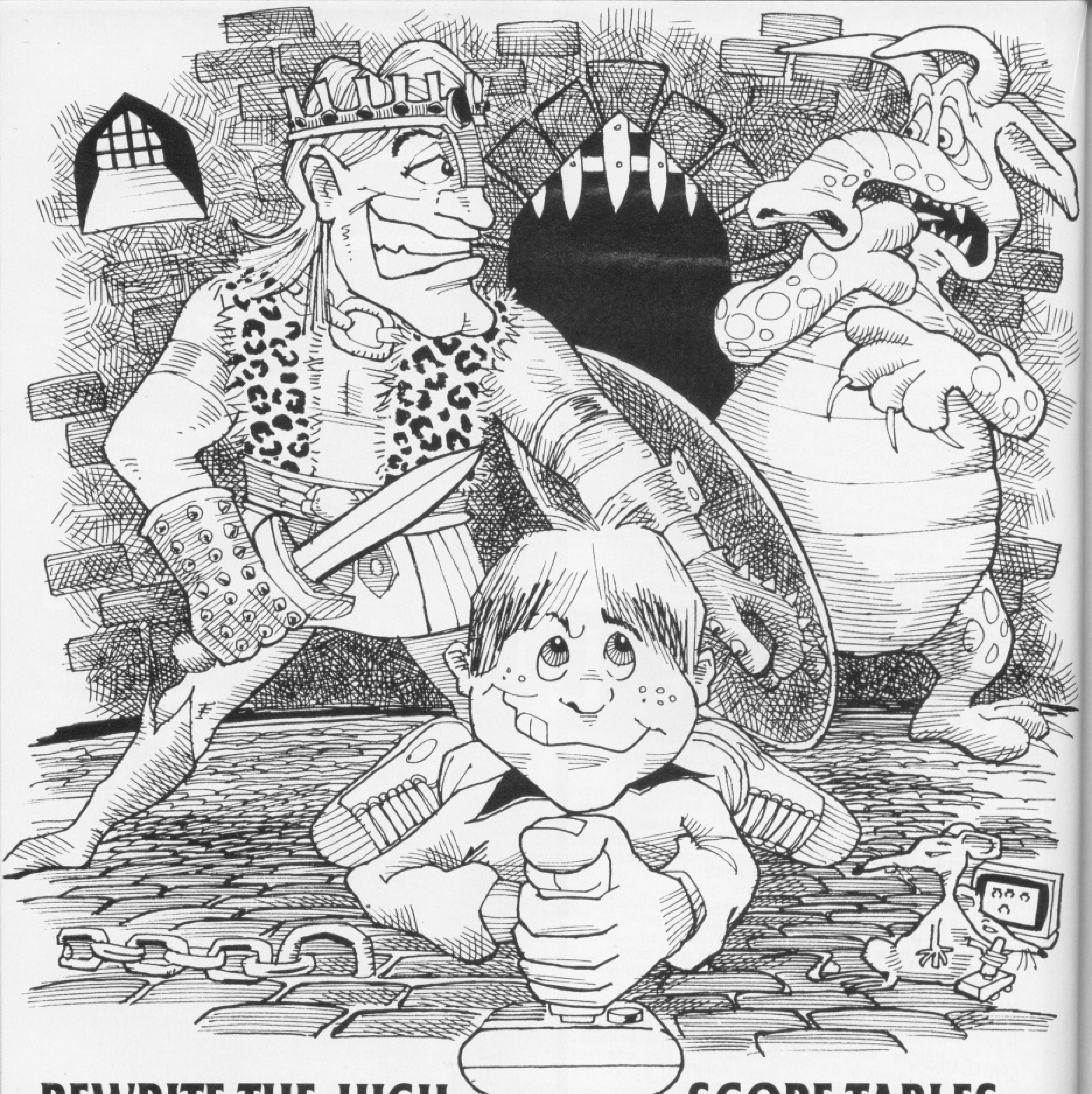
Name: ______

Address: ______

Oasis Software 9a Alexandra Parade, Weston-super-Mare, Avon, BS23 1QT Telephone: (0934) 419921. Every product carries a lifetime guarantee. All prices include extensive manual, VAT and p&p.

24 Hour Access Tele-ordering on (0934) 419921.

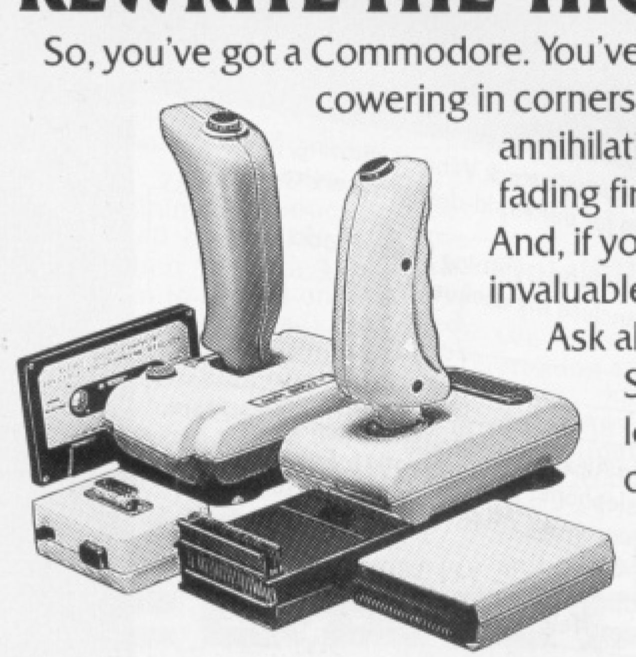
YC 1/85



REWRITE THE HIGH SCORE TABLES

So, you've got a Commodore. You've also got enemies. With the Gunshot, you'll have all the opposition cowering in corners. 8-directional action and an all-in-one moulded stem allows accurate annihilation and strength to survive those all-night sessions. Dual fire buttons for fading fingers (and a rapid fire version when they're really coming thick and fast). And, if you break it (and we know you'll try) our 12-month guarantee will prove invaluable. Only £8.95. The Gunshot plugs directly into the CBM 64 and Vic 20.

Ask about Vulcan interfaces for the C16 and Plus 4.



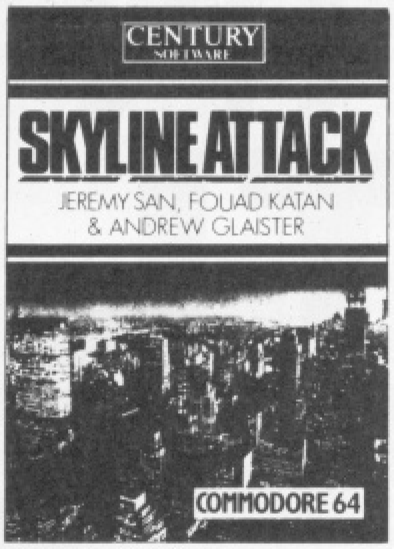
See the range of Vulcan joysticks and interfaces at your local stockist ... we'll see you on the high score tables.

VULCAN ELECTRONICS LTD

Joystick Genius

200 Brent Street, Hendon, London NW4 1BH

CENTURY
COMMUNICATIONS



SKYLINE ATTACK

Pilot your craft across 9 of the world's greatest cities.

This is one of the fastest, most flicker-free games ever written for the Commodore 64.

* **Fabulous animated graphics.** * **18 different types of alien.**
* **Superfast loading.**
* **First program ever to allow you to play another game while it loads!**

£7.95 ISBN 0 7126 0578 9 Commodore 64

ORDER FORM

To: Gill Small, TBS, Church Road, Tiptree, Colchester, Essex CO5 0SR

Please send me............SKYLINE ATTACK cassette(s) at £7.95 (inc VAT and post & packing).

I enclose my cheque/postal order for

£...........(Cheques should be made payable to Tiptree Book Services Limited)

Please debit my *Visa/Access account number
**delete as appropriate*

Account Number

Name

Address

Signed

(Please allow up to 28 days for delivery Subject to availability.)

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema. Most of them comply with the rules contained in the British Code of Advertising Practice.

But some of them break the rules and warrant your complaints.

If you're not sure about which ones they are, however, drop us a line and we'll send you an abridged copy of the Advertising Code. Then, if an advertisement bothers you, you'll be justified in bothering us.

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

TURN TURTLE

Simon Palmer comes out of his shell to look at the Valiant Turtle and the language it speaks, LOGO.

ONE OF THE BIGGEST STEPS so far in educational computing has to be LOGO. It is a language which is very powerful yet simple to use. LOGO is a language which children find easy to grasp, as so many of the fundamental words which make it up are similar or the same as the recognised English definitions. they are designed to achieve. For example, if a child while using LOGO graphics system required the TURTLE on the screen to move forward a set number of steps, he would not have to worry about formulating the BASIC program but simply type 'FORWARD' and then a number of steps. Ease of use is a distinct advantage when teaching a language.

Now, in the past few years a valuable extension to LOGO has been developed. In previous years anyone who owned a TURTLE was either rich or had built it. However, they have become much more freely available, and at a price that can be afforded by the everyday user. The TURTLE is a robot which can be controlled by the computer using LOGO. It is mainly used in conjunction with LOGO's "Turtle Graphics".

When these 'pets' were first introduced they were attached to the computer by an umbilical cord which transmitted the data to the robot. This was a disadvantage as it limited the range of movement. The robot was also bulky and expensive which meant that most schools, could not afford to own one.

Volume manufacturing has since led to a decrease in prices. Technology has improved ten fold and the size of the components has halved. This has lead to some exciting developments such as Valiant Design's TURTLE.

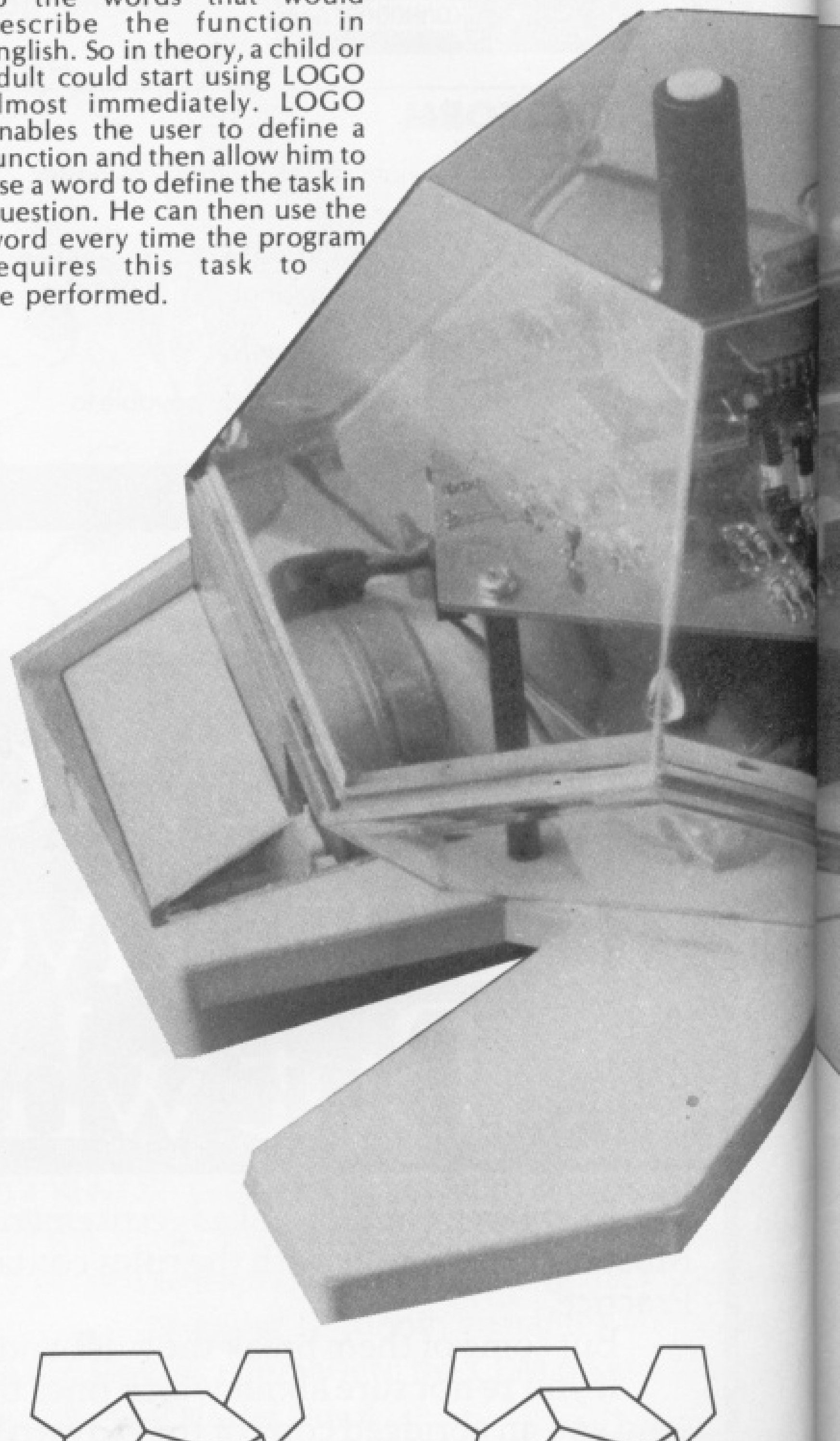
Welcome to the World of LOGO.

LOGO is and has for quite a while, been given little or no acknowledgement of its existence in the U.K. by the general public. A few pockets of activity in schools or computer clubs and the odd article have reminded people of its existence. Perhaps because it doesn't have the software, games or otherwise, it has been unfairly overlooked. But the time has come for people to recognise that it is not just another language but that it is more powerful than BASIC.

LOGO is a procedural language. This means that when a program is generated, it is structured out of procedures. These procedures are either the primitives of the language or have been created to perform a certain function. When a set of procedures are used together, they may call on each other to determine a result as well as performing their own independent tasks.

Speak the same language

In LOGO, the language has been specifically designed to bring both the English Language and the Computer Language closer together to enable use by a greater spectrum of ages. The words used in LOGO are very similar to the words that would describe the function in English. So in theory, a child or adult could start using LOGO almost immediately. LOGO enables the user to define a function and then allow him to use a word to define the task in question. He can then use the word every time the program requires this task to be performed.



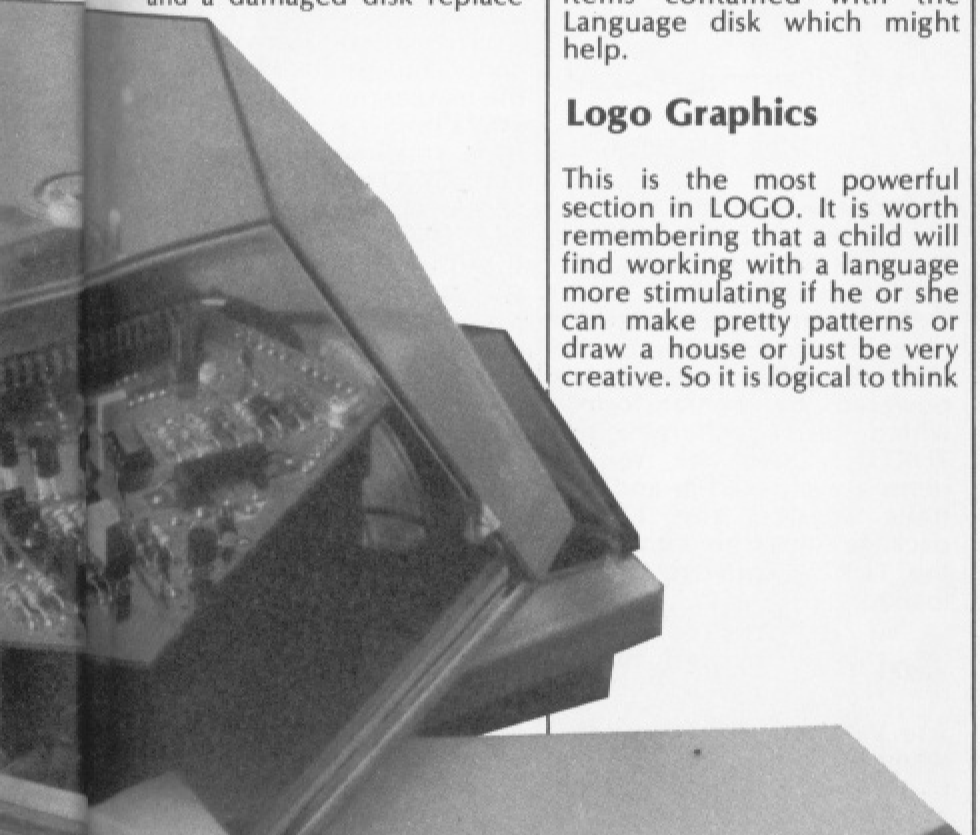


The best way to describe LOGO is as if it were a pyramid. You have the primitives (LOGO itself) at the bottom and on those you build your structure until you have a single word at the top which will perform the whole program, called a SUPER-PROCEDURE.

Commodore Logo

In this plain, small, no-frills box, Commodore supply you with the Language (on disk), the manual which totals well over 300 pages, a utility disk and a damaged disk replacement card. When I thumbed through the manual I found it easy to read and understand, but if I had been a seven or eight year old working alone with LOGO then I would have had difficulties. I would, therefore, suggest that if you are a parent thinking of buying LOGO for your son or daughter, make sure that you can get hold of workbooks for beginners and then use the manual as a reference instead of a tutorial.

After saying that, I must confuse the issue a little by adding that there is a Utility Disk with some fine examples of LOGO graphics and other items contained with the Language disk which might help.



Logo Graphics

This is the most powerful section in LOGO. It is worth remembering that a child will find working with a language more stimulating if he or she can make pretty patterns or draw a house or just be very creative. So it is logical to think that when this language was designed with education in mind, that it would be heavily orientated towards graphics. For example, if I wanted to draw a square in LOGO I would not have to thinkof the various detils of BASIC, I could simply type the following:-

Example 1

```
TO SQUARE
FD  30
RT  90
FT  30
RT  90
FD  30
RT  90
FD  30
RT  90
END
```

or

Example 2

```
TO SQUARE
REPEAT 4(FD  30 RT  360/4)
END
```

Both of the above perform the same task. One is simply an abbreviation of the other.

In Example 1, I gave the procedure the name SQUARE. I then gave the command FD (forward) 30 and RT (right) 90 four times. This made the TURTLE move 30 units forward and turned 90 degrees to the right four times. Example 2 does the same, except I have used a primitive and a different way of getting the angle by dividing the total number of degrees in a circle by the four sides.

So, after playing with the standard graphics via LOGO I decided to move on to higher things. I flicked through my manual until I came to the Sprites chapter. Within this I found that the Utilities disk not only contained examples of how the Sprites could be used but also a Sprite Editor to enable the design of Sprites, always a tricky problem.

For example, if you wanted to move a sprite across the whole of the screen you would have to set the MSB (Most Significant Bit), which is Location 53264. This would, in BASIC, enable a sprite to move on its Xco-ordinate across to the righthand edge of the screen where the register would normally be greater than 255. Well, on LOGO that is already taken care of.

What about sprite to background and sprite to sprite collision? The detection of a Sprite hitting another Sprite is again a complex affair? But on LOGO there are two procedures found on the Utilities disk which enable the computer to gain the necessary data to decide if a collision has occurred. They are "TB" which checks for background collision and "TS?" which checks for Sprite collision. A not so complex problem is getting an enlarged sprite to appear. With BASIC you have two locations 53271 and 53277 which control expansion. Whereas on LOGO, BIGX, SMALLX, SMALLY, will suffice to complete the job.

I was surprised, however, that the manual does not make any reference to Multicolour Sprites. In the 21 pages on Sprites it dealt with everything from defining to saving, but nothing at all on Multicolour mode. It is also worth pointing out that most of the control of Sprites, i.e. the commands, can only be found on the Utilites disk so it is worth making a back-up copy just in case something happens.

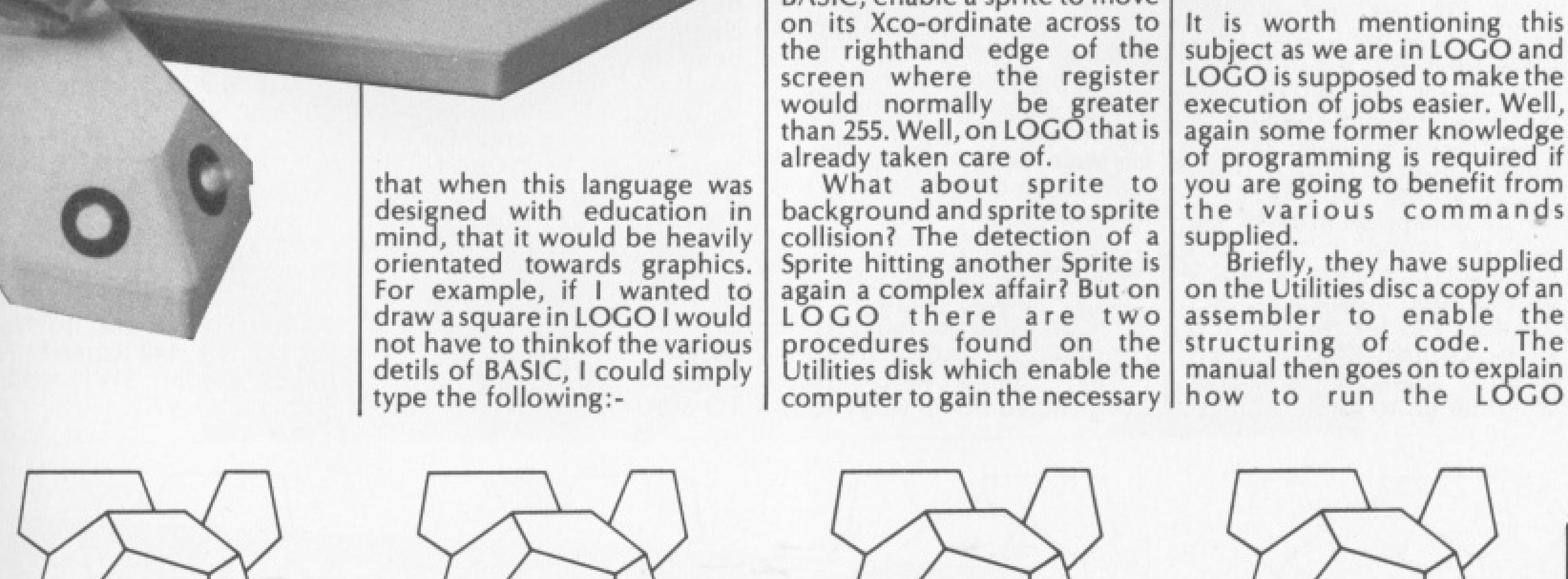
Sound

When I loaded the first file for music on the Utilities disc called Music, I watched it define the various procedures required to gain Sound. I was expecting more commands to be defined but unfortunately only items such as Attack, Decay, Sustain and Release are there. In all fairness, even though the procedures defined for music are very few, they are the important items required for complex music making and obviously, if there isn't enough there, you can create a new procedure by using the various LOGO machine code primitives.

Machine code

It is worth mentioning this subject as we are in LOGO and LOGO is supposed to make the execution of jobs easier. Well, again some former knowledge of programming is required if you are going to benefit from the various commands supplied.

Briefly, they have supplied on the Utilities disc a copy of an assembler to enable the structuring of code. The manual then goes on to explain how to run the LOGO

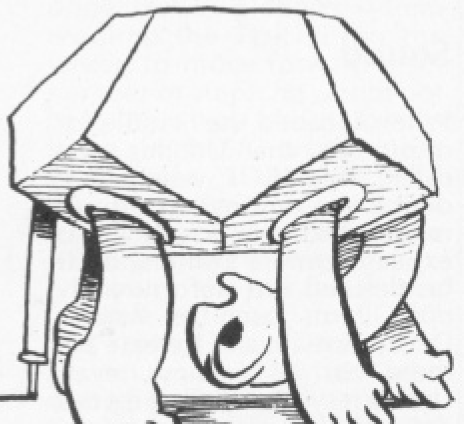


assembler and also gives an example of how to change colours via machine code.

Words, words, words

Words and Lists is the largest section in the LOGO language book even though it covers a relatively small area (17 primitives). Whereas the graphics section would in most cases be of a direct input and result nature, Words and Lists requires indirect operations.

This set of primitives allows you to generate data bases all the way to "intelligent" software. Now I am not going to get into the argument of whether computers can or cannot be intelligent, but with the aid of the primitives the computer can learn and react after a specific action has happened.



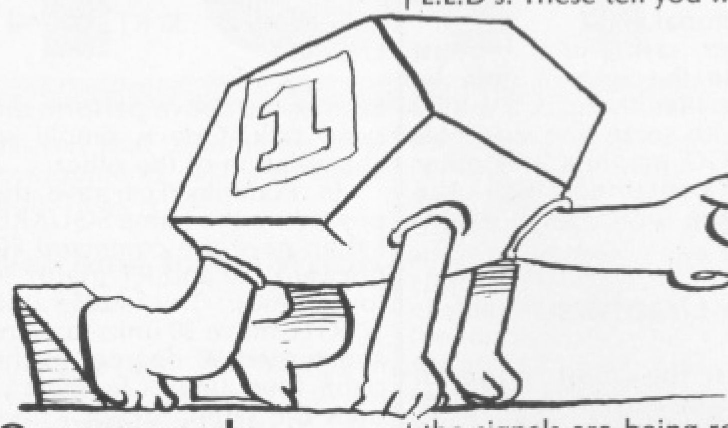
Commodore has supplied a game on the Utilities disc to illustrate this very point.

Finally

Much more on this fascinating subject can be found out by using the package and the TURTLE together or just LOGO on its own. It is worth bearing in mind that even though it is a teaching language, the subtlety of a high-level language has been incorporated. Items which appear within LOGO but do not appear in BASIC can also be found in the more powerful languages available on the open market. So when the user has learned to work with computers he can, without changing languages, continue on to higher things.

The Valiant designs Turtle

The Valiant TURTLE, designed for use with LOGO, is prepared for battle and ready to run as soon as you remove it from the box. It comes complete with an infra-red controller, a power supply/recharger, the various manuals and paperwork, and obviously the software.



On your marks

Before starting to connect the TURTLE I read through the manual. The points which are worth noting here are that the manual can be read by anyone. If they are being taught, it's an aid and if you are teaching it's a guide to enable smooth operations. The manual does contain a large selection of cartoons illustrating the various stages in which the TURTLE can be involved and these highlight very well, any problems that might occur whilst setting the machine up or dûring operation. They also had the foresight to include a large section on trouble-shooting and the various remedies to be administered to any ailing TURTLES.

The one criticism I must make is that even though there is a high-quality magazine called PEN-UP included, I would like to have seen a few example programs contained either with the software or the manual.

Get set

After perusing the paperwork, I started to set up the hardware. Firstly, the TURTLE itself. Constructed out of very robust plastic, the green see-through shell provides a perfect view of the internal workings which have been anchored safely to a cream base. At the front of the TURTLE two light-emitting diodes give an indication of whether the TURTLE is up to strength by maintaining a constant light or by flashing to indicate that recharging isn't far off. On the main board of the TURTLE are two extra L.E.D's. These tell you whether the signals are being received by the TURTLE or whether it is recharging safely.

The infra-red transmitter plugs into the User Port on the CBM 64. The transmitter is powered by a transformer which also recharges the TURTLE. Once the various items are plugged in and you have loaded the LOGO package into the computer, the TURTLE software can be loaded.

Go!

The Valiant TURTLE software, when loaded, redefines some of LOGO's primitives. This makes writing software, which will work with the TURTLE, a little bit difficult as they do not describe what has been changed. Now everything has been set up, you can check the accuracy of your TURTLE. Valiant supply with the TURTLE one marker pen and also one screwdriver. When the TURTLE is ready to work you can adjust the wheels to enable a greater degree of accuracy whilst drawing.

The most efficient way is to get the TURTLE to draw a square.

Example

```
TO SQUARE
REPEAT 4[FD 30 RT 360/4]
```

The above procedure in LOGO should be enough to enable accurate adjustment. The adjustment is made by turning the screws which move the wheels either nearer the TURTLE body or further away.

Turtle on the run

The TURTLE when working moves at a steady speed. One problem I did come across is that the space between the wheels and the floor is very small. This will lead to the TURTLE catching itself and being left high and dry. So make sure it is used on a level surface.

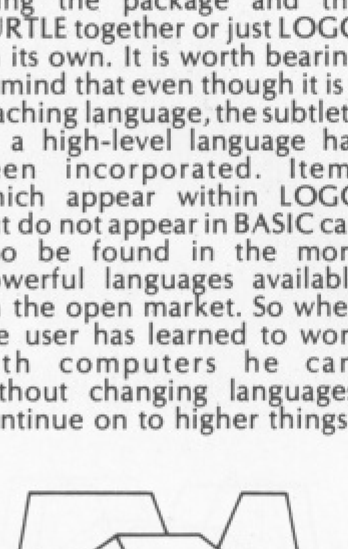
The final item to be connected to the TURTLE was the marker pen. This is inserted into a holder which is serviced by a small servo-motor. The two LOGO commands to raise and lower the pen are PENUP and PENDOWN (PU and PD for short). Make sure that when you write a procedure to incorporate some TURTLE graphics with movement of the robot you use these commands otherwise, like I did at first, you will get more lines than you bargained for.

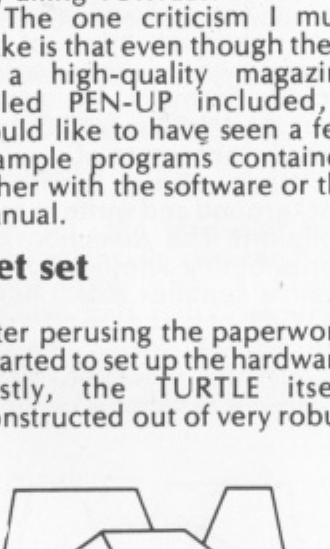
From start to finish

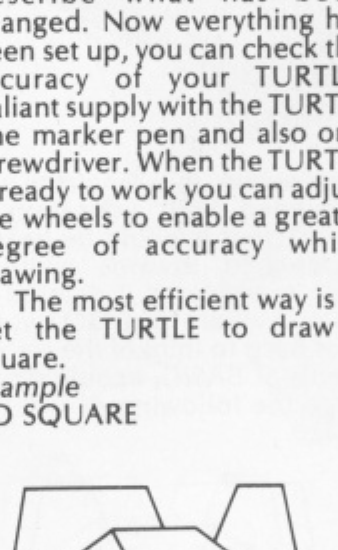
I found the TURTLE a pleasure to use; it is very easy to set up and even easier to operate. Commands such as TURTLE which will direct the input from the computer to the TURTLE or, if the command is repeated, will stop the signals are very useful. The operation of the TURTLE can be speeded up marginally by hiding the computer's sprite version of the TURTLE. This means there is less for the computer to operate.

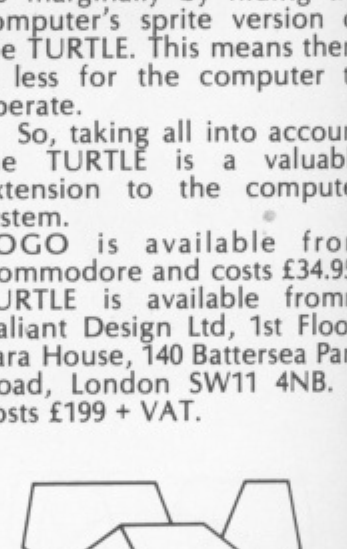
So, taking all into account the TURTLE is a valuable extension to the computer system.

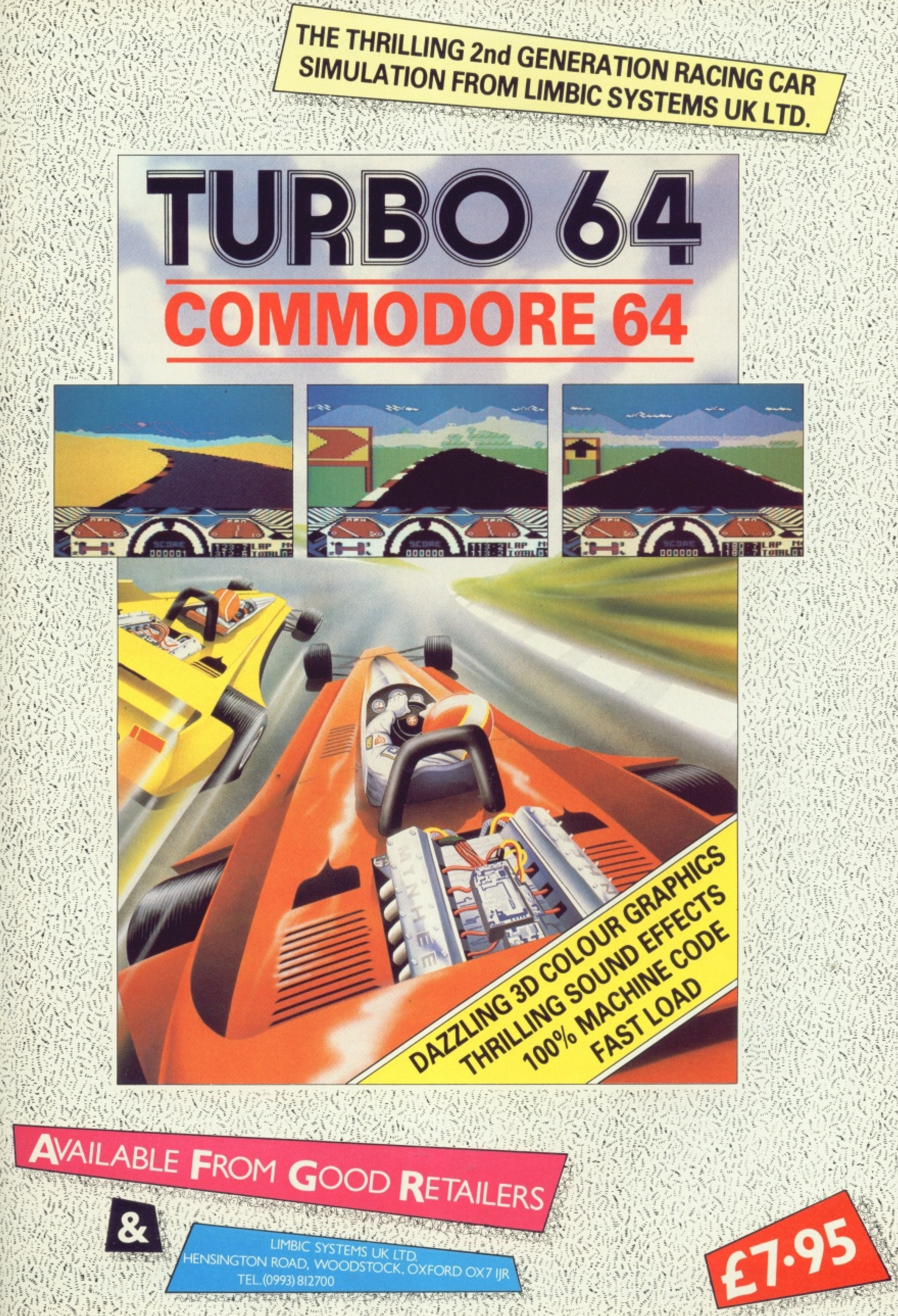
LOGO is available from Commodore and costs £34.95. TURTLE is available fromm Valiant Design Ltd, 1st Floor, Para House, 140 Battersea Park Road, London SW11 4NB. It costs £199 + VAT.










THE THRILLING 2nd GENERATION RACING CAR SIMULATION FROM LIMBIC SYSTEMS UK LTD.
TURBO 64
COMMODORE 64
DAZZLING 3D COLOUR GRAPHICS
THRILLING SOUND EFFECTS
100% MACHINE CODE
FAST LOAD
AVAILABLE FROM GOOD RETAILERS
&
LIMBIC SYSTEMS UK LTD.
HENSINGTON ROAD, WOODSTOCK, OXFORD OX7 1JR
TEL. (0993) 812700
£7·95

LIGHT FANTASTIC

Dave Crisp throws some light upon a couple of the more popular lightpens on the market.

A COUPLE OF YEARS AGO IT WAS ONLY possible to see lightpens being used on the most expensive computers. Between now and then the lightpen has been made to seem old hat. Special monitors now give the user the ability to simply touch the screen with their finger to make a decision relevant to the program, but it is going to be a long time before this system is available at a reasonable enough price to implement in the home.

Until then one thing that can take the place of the keyboard in some respects is a lightpen. It is called a lightpen because of its shape and the fact that it responds to light emitted from the TV or monitor screen. A lead connects the pen to a port on the computer and it is linked to the main program by short subroutines which are able to detect the pen's position on the screen. It is able to do this because the monitor screen is scanned at a particular rate. When the lightpen is pointed at the screen and a switch is pressed, the decoding software calculates the position of the pen from the point at which the scan is detected.

In older pens ambient light often made reading the position of the pen a very hit and miss affair but, with improved circuitry, it is possible for the pen to react only to light from the monitor. The main use for lightpens on home micros seems to be for drawing and design programs. Software detects the position of the pen on the screen and as you move the pen a line may be drawn or the position of the pen may be a reference point for the start of a line, circle or triangle etc. It is also possible to indicate the part of a drawing which is to be filled with colour. All these things could be done with the cursor keys but in most cases the use of a lightpen speeds up positioning and improves accuracy.

Another use for a lightpen is in programs where a lot of decisions are to be made. Since I've had the Stack lightpen I have been able to write a couple of educational games where my youngest daughter can complete the whole program without using the keyboard. There is no need for her to be able to recognise letters or keys as the options can be displayed on the screen in a graphic form and the decisions can be made from this.

It is usually fairly easy to adapt home written software to work with a lightpen but it is also easy to go overboard with the pen. If the user continually has to go from the lightpen to the keyboard and back again it becomes as much a trial as a benefit. I have not seen much software available to use with lightpens in a useful way and, as I am a nurse by trade and have a couple of patients who have lost most of the use in their arms, I feel that it should be possible for me to write software where they are able to use a computer in many ways using only the lightpen, possibly in their mouths to amuse, educate and help them.



Datapen lightpen (price: £25.00)

The **Datapen Lightpen** is easy to use and comes with three programs to get you started. It is plugged into one of the joystick ports and the lead is long enough to enable you to reach all the screen. Although that sounds obvious I have seen a lightpen where it was only possible to reach one side of the screen. The software with the pen (tape based) can be saved easily to disc, and the first program simply tells a little about the program and enables you to move sprite graphics around the screen. It also shows the way movement of the pen effects the variable values used in the detection of the pen point. The other two programs are both for drawing on screen. The first allows low-res block graphics in all the colours on the 64 to be drawn on the screen, the second allows drawing in high-res mode. This mode is single colour but with practice neat little drawings could be achieved and then save to tape. In order to save the disc it was first necessary to slightly adapt the program. My young children enjoyed the two graphic programs and soon they became quite proficient. The manual that comes with the pen is of a fairly low quality but it is packed with useful information which makes using the lightpen in your own programs very easy indeed.

Quality of finish

When you use it in your own software the switch on the pen can be programmed in such a way as to make the pen inoperative until the point is at the desired location or to perform a particular operation when it is pressed. This is a feature that was missing on some earlier pens and it made them difficult to use. The quality of finish on the pen was not fantastic compared to the Stack lightpen but it was robust. The hole at the back of the pen from which the LED sprouted seemed to have been cut out with a hammer and chisel and the moulding of the body was rather spoiled by a large amount of flash. Small points these but it could have been finished a little better. The LED is there to indicate that the pen is receiving some sort of signal and gives out a reassuring glow when plugged in. When the switch is pressed it glows brighter and serves to indicate as a visual reminder that the position has been calculated. I enjoyed using this pen and I am sure that, given the time, I shall be able to use it to its full potential. It can be made into a useful item as much at home, in business and education, as in games.

Stack Lightpen (Price: £28.00)

Later I received the **Stack Lightpen**. The pen comes in a fairly stout box along with a manual and tape based software. With my SX-64 the first thing I had to do was to get the software onto disc. Fairly clear instructions are given in the manual on saving the software to disc with a monitor. However you can run into trouble. Most of the software that comes with the pen can be loaded and saved as the programs are all BASIC. The better games and the Paintbox software are in machine code and is in several parts. As I said the documentation tells you how to save this to disc with the aid of a monitor. I have three monitors and I could not do it with any of them as they conflicted in memory with the software. I eventually managed to save it onto disc.

Playing the games was not exactly mind blowing and had I bought them I would have been disappointed. However they are free and you don't look a gift horse in the mouth!

Clever these Liverpudlians

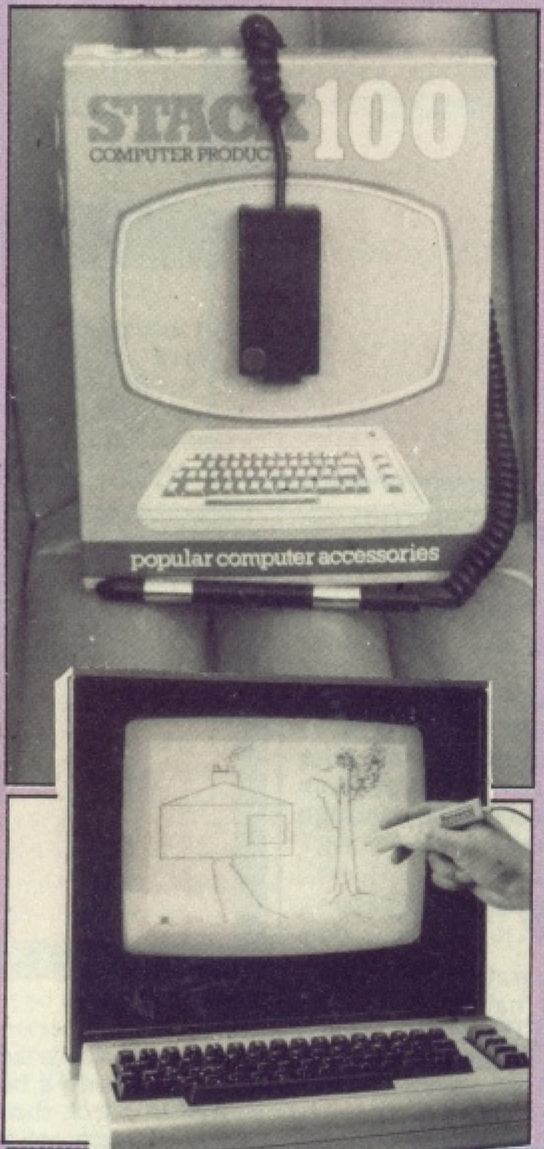
The accuracy of the lightpen amazed me. I was lucky enough to talk to a young lad from *Stack* a few weeks ago at a trade show and he took a lot of trouble to explain to me why the pen was so accurate and reliable. Many pens simply read the position at which the pen is pointed and plot that position. The Stack pen however, takes sixteen readings, excludes the four highest and lowest numbers and then calculates the mean average of the remaining eight: very clever and very reliable. The end result of this is that if you try to draw a straight line freehand then you usually manage to. No real need to do that though as it is possible to LOCK the TRACK of the pen either horizontally or vertically and draw dead straight lines with ease. For diagonal straight lines you simply PLOT a point where you wish your line to start, move the pen to the point where you wish your line to end and press the up arrow key. Bang! a straight line. A similar method is used to draw squares and rectangles. Plot one corner, move the pen to a position where you would like the opposing corner to be, press 'Q' and there you have it. Circles were not quite as successful as the other shapes but a similar method is employed and you end up with a slightly squashed circle.

Unlike a few

Some machines when used with light pens and drawing aids do not like you drawing over the screen edge and often come up with an error report. The Stack pen however allows you to draw over the edge. Of course nothing is drawn on the screen border, but the line or shape will at least go up to the edge. I did find, however, that when this occurred there were odd times when you would get wrap-round and the rest of the line would enter the screen at the opposite edge — Not a serious problem though.

Finishing touches

The Paintbox program had some fantastic little touches. To my mind the most impressive was that on one screen you could have a set of diagramatic designs, and you could 'PICK UP' the desired design, 'CARRY' it onto your work screen and DROP it where you wanted: excellent for planning etc. You are provided with electrical component designs and house part designs on tape and that saves you having to draw them. Another extra is the ability to ZOOM into a small section of your work sheet. Using this method it is possible to plot pixel by pixel and also UNPLOT. This made it possible to do very precise work and turned the lightpen from a plaything to a fairly serious design tool.



Pictures to hang

With a Commodore printer it was easy to get a screen dump of any masterpieces you had created (if you can stand the wait) and also using the Stack Centronics interface with another printer.

Ready to improve

Talking to Stack, I learned one or two snippets of information. Improvements are underway and one or two little niggles I pointed out appear to have been noted already and acted upon. the most frustrating of these was the ease with which it was possible to lose your picture. One press of the 'c' key and all was gone. I lost work on more than one occasion and turned the air blue. Apparently work is being done on it.

Grizzle number two is the switch on the pen. It comprises two bands of metal — one near the end of the pen and one near the top. Unless you have very large rubber hands it is not easy to put your hand on both contacts at once (to activate the switch). This made use of the pen by children very difficult and I soon noticed that, very often, my five year old daughter was using two hands to do it. Apparently this too has been noticed by Stack and I believe that later they will have the two bands closer together at the end of the pen.

First class

Almost everything about this pen was well above average. The quality of finish, the quality of the materials used, the actual Paintbox software, and so on. Only the quality of the games provided and the little faults mentioned above scratched it's highly polished finish. One thing I must point out is that it is not a multi-colour paintbox. It is possible to globally change the colours of screen, ink, and border but it is not possible to have more than one ink or screen colour displayed at once, although I didn't find this too great a problem.

Below is a quick list of its commands so you should be able to see what a well thought out piece of software it is.

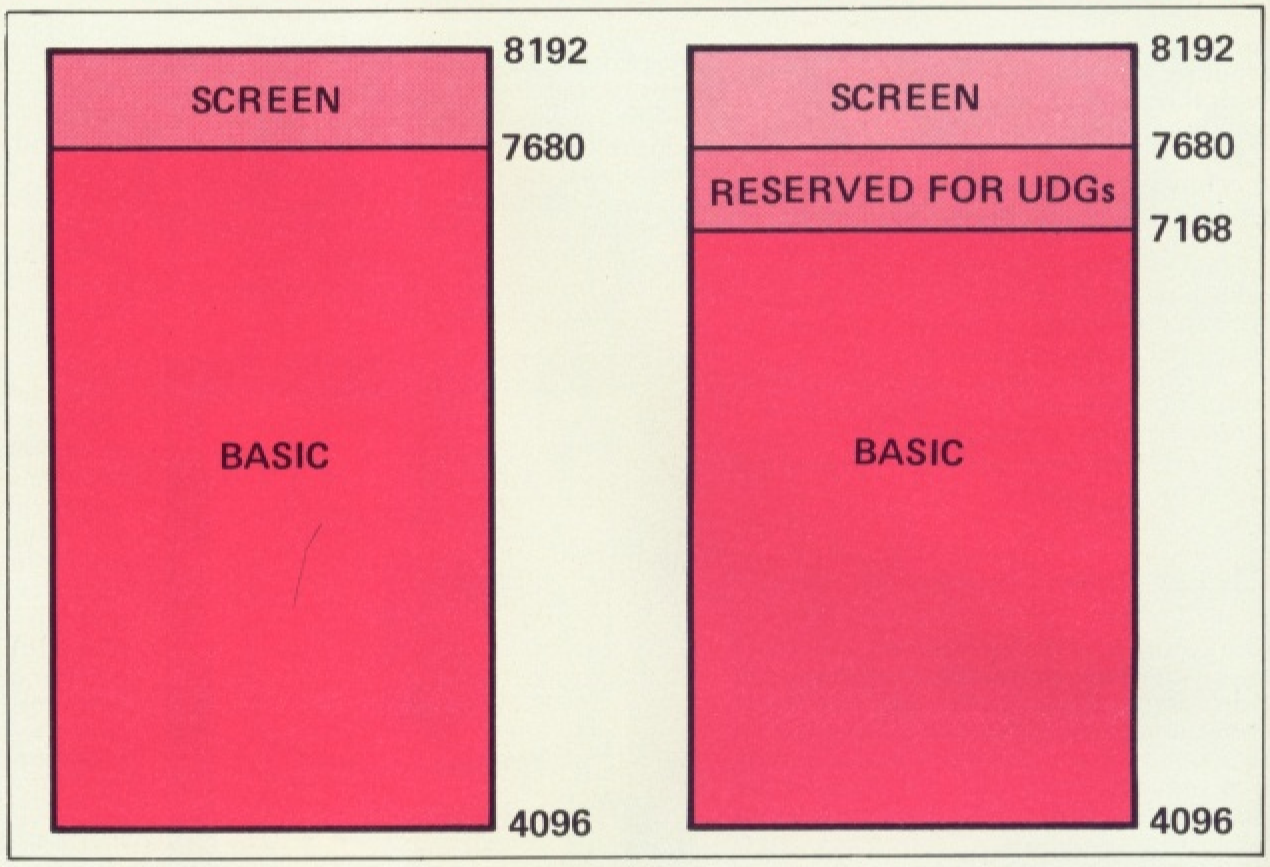
For those of you who want a multicolour paintbox all I am able to say is 'WAIT': Stack have some cards still to play.

MERGE	merge the two screens together
CIRCLE	draw a slightly squashed circle
QUADRIL	draw square/rectangle
GET	pick up a small area of the page
DROP	use to leave above on desired spot
CLEAR	clear screen (whether by accident or design)
FILL	fill an enclosed area with present ink colour
INVERSE	invert screen colours
KEYS 8	lock pen's travel horizontally
KEY 9	lock pen's travel vertically
READ	retrieve a picture from tape/disc
WRITE	save a picture to tape disc

This table does not contain a complete list of commands but it includes the main points of interest.

This month Bryn Phillips examines the creation of User Defined Graphics on your VIC 20.

VIC GAMES PROGRAMMING



1a and 1b

THIS IS THE FOURTH OF A five part series of BASIC Games Programming for the VIC 20. The series is primarily intended for newcomers to games programming, but there might well be a few useful tips for seasoned programmers.

The VIC 20 has a very well thought out graphics set which can be used to good effect in many games. However, if you want a single character to represent an object such as a dog, an alien or a spaceship it's really stretching the imagination too far to stick to the standard graphics set. To get a professional finish to your game you need to define your own characters. The steps involved in creating User Defined Graphics (UDGs) are outlined below:

1) Make room in the memory for the graphic set.
2) Calculate the data for the UDGs.
3) Put the UDGs into the reserved memory and direct the VIC 20 to use the UDGs.

It sounds a lot of work, but and in practice it's quite straightforward. A few POKEs and Data statements, and you're there — but let's start at the beginning.

Making Room for the UDGs

The standard character set supplied with the VIC 20 is stored in ROM, and it can't be altered. If you want your own customised graphic set you have to put it in RAM. The way you do this depends on what expansion you have on board. This month we will be considering the UNEXPANDED VIC 20, and I will go over the procedure for dealing with expansion in the next article. The memory map for the unexpanded VIC 20 is shown in fig 1a. You can alter the top of BASIC memory pointer to 7167 as shown in fig 1b. Switch on your VIC 20 and enter:

POKE52,28:POKE56,28:CLR

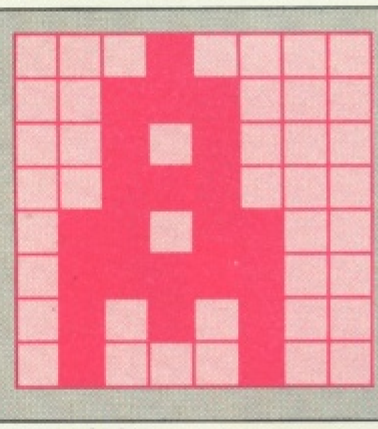
That's it. You've changed the memory map. Now enter:

PRINT FRE(0)

It's gone down to 3069. You've lost some BASIC, but you've gained enough space for your UDGs. In fact you've got room for 64 — that should be enough for most games.

Designing the UDGs

Each VIC character is made up from an 8x8 matrix of pixels, as illustrated by the rocket in Fig 2. The rocket is made up of 8 rows, each of which can be represented by a binary number as shown in Fig 2b. The binary number is obtained simply by designating each pixel which is switched on as 1, and each pixel which is switched off as 0. The binary number is then converted into decimal as shown in Fig 2c.



0	0	0	1	0	0	0	0
0	0	1	1	1	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	0	0	0
0	1	1	0	1	1	0	0
0	1	1	1	1	1	0	0
0	1	0	1	0	1	0	0
0	1	0	0	0	1	0	0

2a and 2b

The process is simplified by starting off with a blank grid as shown in Fig 3, shading in the required squares, and adding up the numbers shown at the top of each column to give the decimal value of each row.

We're almost there now. All that is left is to POKE the character data into the allocated memory locations.

Listing 1

```
READY.

 1 REM LISTING 1
 2 :
 10 POKE52,28:POKE56,28:CLR
 20 FORL=7432TO7439:READD:POKEL,D:NEXT
 30 DATA16,56,40,56,108,124,84,68
```

Entering the UDG data

Having re-defined the VIC memory, the characters are

Figure 2c

2^4	= 16
$2^5+2^4+2^3$	= 56
2^5+2^3	= 40
$2^5+2^4+2^3$	= 56
$2^6+2^5+2^3+2^2$	= 108
$2^6+2^5+2^4+2^3+2^2$	= 124
$2^6+2^4+2^2$	= 84
2^6+2^2	= 68

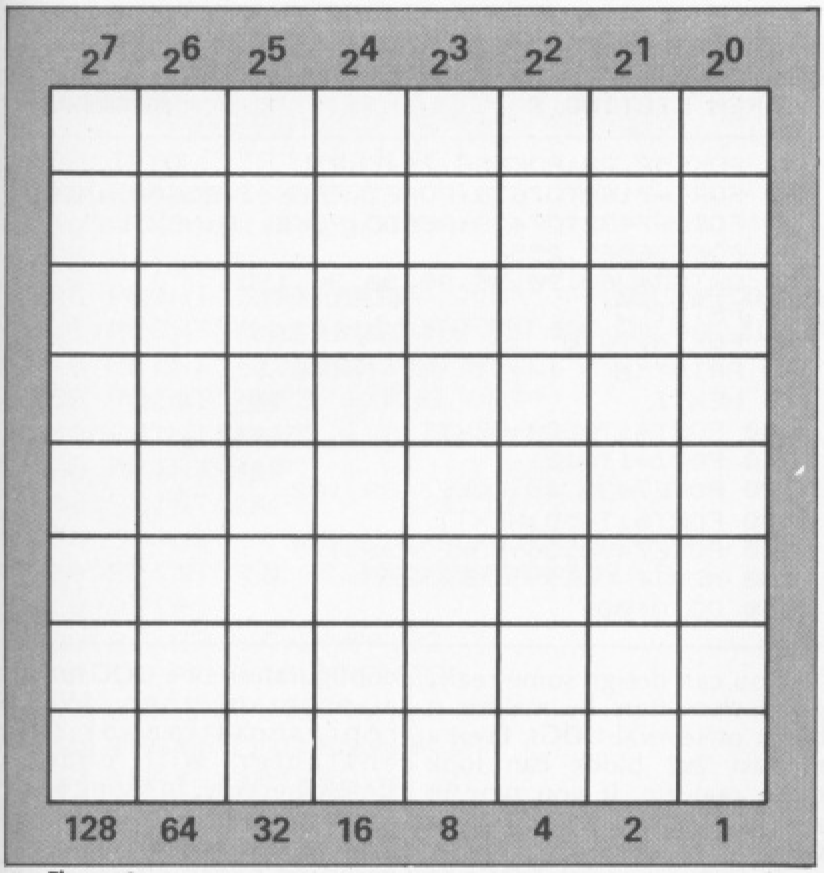


Figure 3

read from 7168 onwards. Each character is made up of 8 bytes and, so the first character is stored at 7168-7175, the second at 7176-7183, and so on as shown in Fig 4:

mess! All the characters have been converted to garbage apart from the "!". Press this key and you've got the rocket. So far so good — only 63 more characters to go!

Character	Display Code	Memory Location
	0	6168-7175
A	1	7176-7183
B	2	7184-7191

Figure 4

The start and finish address of each character is easily calculated using the Display Code (CD).

Start Address = 7168 + CD★8
Finish Address = 7168 + CD★8 + 7

Now to enter our UDG. Let's use character number 33, the "!". The start address is 7432, and the finish address is 7439. The data can be POKEd in quite simply using Listing 1. Now enter:

POKE 36869,255

This instructs the VIC 20 to get its character information from RAM starting at 7168. What a

It's unlikely that you will want to re-define the entire character set as you'll probably want quite a few of the standard characters in your game. Fortunately the solution is quite simple. All the data you need for these characters is already in ROM, and all you have to do is to copy these into RAM, a painless operation, and then just change the ones you want. Press RUN/STOP and RESTORE (back to normal) and enter and RUN Listing 2. Line 20 copies out the character set, and line 30 modifies the character. If you want to modify more than one character you just change the start and finish address in line 30 or, if the characters you want to change are not consecutive, add additional lines as necessary, following the format of line 30.

Listing 2

```
1 REM LISTING 2
2 :
10 POKE52,28:POKE56,28:CLR
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
30 FORL=7432TO7439:READD:POKEL,D:NEXT
40 POKE36869,255
50 DATA16,56,40,56,108,124,84,68
```

Making it Easy

So far the whole business seems to be pretty laborious, particularly calculating the eight bytes required for each character. Writing games should be fun, and whilst working out lots of binary to decimal conversions might improve your mental arithmetic, it tends to knock your inspiration for six. Besides that, when you try your character out on the screen for the first time it probably won't be exactly what you want, and you're back to the drawing-board! A far better way of designing your UDGs is to do the job directly on the screen. The utility program in this article, "CHARACTER DESIGNER", will allow you to do this.

On RUNning the program you are presented with a blank 8x8 grid, which you can fill in using the cursor control and function keys, as directed in the program. The actual character is shown to the right of the grid, and the byte values for this character are continuously updated on the right of the grid. If you have a 1520 printer plotter, lines 980-1310 will give a hard copy reproduction of the character and the values for the data statement. An example of the output obtained from the 1520 is shown in Fig 5. If you don't possess one of these printer plotters, just miss out this part of the program — it's not essential.

Listing 3

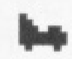
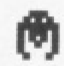
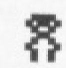
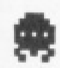
```
1 REM LISTING 3
2 :
10 POKE52,28:POKE56,28:CLR
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
30 FORL=7448TO7479:READD:POKEL,D:NEXT
40 POKE36869,255
50 DATA0,0,1,1,7,15,15,15,0,0,129,255,225;240,252,240
60 DATA63,127,127,127,127,63,42,63,252,254,254,254,254,252,170,252
70 PRINT"[illegible]#$"
80 PRINT"[illegible]%&"
```

Listing 4

```
READY.

1 REM LISTING 4
2 :
10 POKE52,28:POKE56,28:CLR
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
30 FORL=7432TO7455:READD:POKEL,D:NEXT
40 :
50 REM CHARACTER DATA
60 :
70 POKE36869,255
80 DATA96,96,80,96,64,160,192,128
90 DATA48,48,32,112,104,48,40,68
100 :
110 REM CHARACTER CODES !,",#
120 :
130 DATA3,3,2,14,21,6,9,9
140 FORI=1TO3:READC(I):NEXT
150 DATA33,34,35
160 P1=8164:P2=38884
170 PRINT"[illegible]"
180 FORX=1TO20
190 FORI=1TO3
200 POKEP1+X-220,C(I):POKEP2+X-220,0
210 FORJ=1TO100:NEXT
220 POKEP1+X-220,32
230 NEXTI,X
240 GOTO180
```

Figure 5

	126	129	219	153
	129	189	165	126
	0	128	192	230
	255	255	66	0
	56	84	254	186
	170	146	130	68
	60	90	36	24
	36	66	36	36
	56	124	254	170
	254	124	146	68

Listing 5

```
1 REM LISTING 5
2 :
10 DIMCH(2,2)
20 POKE52,28:POKE56,28:CLR
30 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
40 FORL=7432TO7471:READD:POKEL,D:NEXTL
50 POKE36869,255
60 DATA16,16,16,16,16,16,16,16
70 DATA 0,0,0,128,96,128,0,0
80 DATA0,0,0,8,6,8,0,0
90 DATA16,16,16,144,112,144,16,16
100 DATA16,16,16,24,22,24,16,16
110 FORX=1TO2
120 FORY=1TO2
130 READCH(X,Y)
140 NEXTY,X
150 DATA34,35,36,37
160 PRINT"■"
170 FORI=1TO21
180 PRINT"■! ! ! ! ! ! ! ! ! ! ! !"
190 NEXTI
200 P1=8164:P2=38884
210 FORX=1TO21
220 PK=PEEK(P1+X-220)
230 I=1:IFPK=33THENI=2
240 POKEP1+X-220,CH(I,1):POKEP2+X-220,0
250 FORJ=1TO50:NEXTJ
260 POKEP1+X-220,CH(I,2)
270 FORJ=1TO50:NEXTJ
280 POKEP1+X-220,PK
290 NEXTX
300 GOTO210
```

Listing 6

```
READY.

1 REM LISTING 6
2 :
10 POKE52,28:POKE56,28:CLR
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
30 FORL=7432TO7439:READD:POKEL,D:NEXTL
40 POKE36869,255
50 DATA24,60,90,36,90,90,36,195
60 PRINT"■"
70 FORI=1TO10
80 PRINT"■ ! ! ! ! ! ! ! ! ! ! !"
90 NEXTI
100 FORT=1TO500:NEXTT
110 FORI=1TO10
120 POKE7438,66:POKE7439,102
130 FORT=1TO50:NEXTT
140 POKE7438,36:POKE7439,195
150 NEXTI
160 GOTO100
```

Imaginative use of UDGs

To finish off this month, it's worth thinking about some of the things we can do with the UDGs, apart from simply regarding them as little objects which we can move around the screen. There's enormous scope, and if you're not careful you'll stray into the realms of computer art and forget all about that game you want to write!

You can design some really good characters by making a block of several UDGs. Even a modest 2x2 block can look quite realistic. If you type in and RUN Listing 3 you'll get a tank.

By using several UDGs to represent an object you can get reasonable animation. You can really make a ball look as though it's bouncing by squashing it as it hits a wall, or you can make a funny little man run across the screen. Try Listing 4 — you'll see what I mean. By working really hard you can simulate sprite movement. Enter and RUN Listing 5, and work out how it's done.

Finally, you can use UDGs to give you lots of movement quickly. If you alter the configuration of a UDG during a program then EVERY representation of that character will change simultaneously. In listing 6, 100 aliens waggle their legs simultaneously. Now that really could be put to effective use. Couldn't it?!

Next Time

If you've followed this series through you should now have all the tools at your disposal to write stunning and effective games in BASIC on the UNEXPANDED VIC 20.

Next month I'll cover memory expansion, and go through some of the techniques which can be used to give a bit of polish to your programs.

Program Listing

```
READY.

10 REM CHARACTER DESIGNER
20 REM BRYN PHILLIPS 1984
30 POKE36879,29:PRINT"■■■                                         "
40 PRINT"■■  CHARACTER DEFINER    "
50 PRINT"■■                       "
60 PRINT"■■ THIS PROGRAM WILL      ALLOW YOU TO DESIGN    YOUR OWN   CHARACTERS"
70 PRINT" AND WILL PROVIDE YOU  WITH THE CODE FOR THE DATA STATEMENTS."
80 PRINT"■■ USE THE CURSOR KEYS    FOR DESIGNING THE      GRID."
90 PRINT"■  ■■PRINT OUT VIA 1520"
100 POKE52,28:POKE56,28:CLR
110 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
120 FORC=7384TO7399:READA:POKEC,A:NEXT
130 DATA 255,255,255,255,255,255,255,255,170,85,170,85,170,85,170,85
140 PRINT"■■■■ ■                   "
150 PRINT"■ ■ PRESS S TO START  "
160 PRINT"■ ■                   "
170 GET A$
```

Program Listing

```
180 IFA$<>"S"THEN170
190 POKE36869,255
200 P1=7707:P2=38427
210 X=1:Y=1
220 PRINT"███████\\\\\\\\\\\\\\\\\\\\\\\\█\█\█\█\█\█\█\█\█\"
230 PRINT"██████\█\█\█\█\█\█\█\█\\\\\\\\\\\\\\\\\\\\\"
240 PRINT"█████F1 = FILL IN"
250 PRINT"██F3 = RUB OUT"
260 PRINT"██F5 = CLEAR GRID"
270 PRINT"██F7 = PRINT OUT"
280 PRINT"██F8 = TERMINATE"
290 GOSUB760:POKE7808,0:POKE38528,0
300 PRINT"██DATA████████████U.D.G."
310 GETA$
320 POKEP1+X+22*Y,32
330 IF PK<>27 THEN350
340 POKEP1+X+22*Y,27
350 IFA$="█"THENY=Y-1
360 IFA$="█"THENY=Y+1
370 IFA$="█"THENX=X-1
380 IFA$="█"THENX=X+1
390 IFA$="█"THENGOSUB560:REM F1
400 IFA$="█"THENGOSUB660:REM F3
410 IFA$="█"THENGOSUB760:REM F5
420 IFA$="█"THENGOSUB980:REM F7
430 IFA$="█"THENPOKE36869,240:PRINT"█":END:REMF8
440 IFX<1THENX=1
450 IFX>8THENX=8
460 IFY<1THENY=1
470 IFY>8THENY=8
480 PK=PEEK(P1+X+22*Y)
490 IFPK=27THEN520
500 POKEP1+X+22*Y,28
510 POKEP2+X+22*Y,2
520 GOSUB860:GOTO310
530 :
540 REM FILL IN
550 :
560 POKEP1+X+22*Y,27
570 POKEP2+X+22*Y,0
580 ML=7167+Y
590 BX=2↑(8-X)
600 VL=PEEK(ML)ORBX
610 POKEML,VL
620 RETURN
630 :
640 REM RUB OUT
650 :
660 POKEP1+X+22*Y,28
670 POKEP2+X+22*Y,2
680 ML=7167+Y
690 BX=2↑(8-X)
700 VL=PEEK(ML)-(PEEK(ML)ANDBX)
710 POKEML,VL
720 RETURN
730 :
740 REM CLEAR GRID
750 :
760 FORY=1TO8
770 FORX=1TO8
780 POKEP1+X+22*Y,32
790 POKE7167+Y,0
800 NEXTX,Y
810 X=1:Y=1
820 RETURN
830 :
840 REM DISPLAY DATA
850 :
860 PRINT"██"
870 FORI=7168TO7175
880 C$=STR$(PEEK(I))
890 C=LEN(C$)
900 IFC=4THENPRINTC$
910 IFC=3THENPRINT" "C$
920 IFC=2THENPRINT"  "C$
930 NEXTI
940 RETURN
950 :
960 REM SET PRINTER
970 :
980 OPEN1,6,1:OPEN4,6
990 PRINT#1,"M",0,-50
1000 PRINT#4
1010 PRINT#1,"M",0,-200:
     PRINT#1,"M",0,0
1020 :
1030 REM DRAW CHARACTER
1040 :
1050 FORX=1TO8:FORY=1TO8
1060 PK=PEEK(P1+X+22*Y)
1070 IFPK=32THEN1180
1080 PRINT#1,"M",5*X,-5*Y
1090 PRINT#1,"I"
1100 FORA=0TO5
1110 PRINT#1,"R",A,0
1120 PRINT#1,"J",A,5
1130 NEXTA
1140 FORA=0TO5
1150 PRINT#1,"R",0,A
1160 PRINT#1,"J",5.A
1170 NEXTA
1180 NEXTY,X
1190 FORI=1TO4
1200 J=7167+I
1210 PRINT#1,"M",I*80,-15
1220 PRINT#4,PEEK(J);
1230 NEXTI
1240 FORI=5TO8
1250 J=7167+I
1260 PRINT#1,"M",(I-4)*80,-40
1270 PRINT#4,PEEK(J);
1280 NEXTI
1290 CLOSE1:CLOSE4
1300 X=1:Y=1
1310 RETURN
```



ALL CLEAR



Owen Manderfield shows you how to clarify control characters by inserting 'REM' lines into BASIC programs.

HAVE YOU EVEN TRIED typing in listings of BASIC programs for your Commodore 64 involving long lines or cursor controls and colour commands? If so, you will probably know that deciphering the graphic symbols representing these controls can be very difficult. This is a real problem for programmers producing programs for publication in magazines such as this one, as the program will have to be typed in from a printed listing. One way of overcoming this problem is to document your programs with 'REM' statements, explaining the control characters used.

Here is a simple machine code routine which inserts 'REM' lines in any BASIC program held in memory immediately before every line that contains the line:

```
1020 PRINT"■■HELLO":?
```

Then a line 1019 is inserted showing that a 'cursor right' and a 'cursor down' character are to be printed.

```
1019 REM CUD-CUR
1020 PRINT"■■ IELLO":?
```

The dash between the two commands shows that the control characters are next to each other. If they are separated, then a space is printed, e.g.:

```
1019 REM CUR-CUD-RED

          ■THERE ■■■YOU!■
```

This means that 'HELLO' is printed in red, 'THERE' in black and 'YOU' in reversed green. Function keys can also be shown:

```
1059 REM F1  F2
1060 IFA$="■"ORA$="■"
```

Program Listing

```
10 READP:P1=P:TT=0:LN=1000:F=0
20 CS=0:LN=LN+10:PRINT"READING"LN"]"
30 READA$,A:IFA$="END"THEN90
40 FORI=1TOLEN(A$)STEP2:H=ASC(MID$(A$,I))-48:IFH>9THENH=H-7
50 L=ASC(MID$(A$,I+1))-48:IFL>9THENL=L-7
60 N=H*16+L:CS=CS+N:POKEP,N:P=P+1
70 NEXT:IFCS<>ATHENPRINT"**** ERROR IN LINE"LN"****":F=1
80 TT=TT+A:GOTO20
90 IFTT<>ATHENPRINT"**** LINE COUNT ERROR ****":F=1
100 IFF=0THENPRINT"ALL DATA OK.  PROG ="P1"TO"P
1000 DATA  49152
1010 DATA A52B8560A52C8561, 876
1020 DATA A000B1608562C8B1, 1041
1030 DATA 6085630562D0034C, 718
1040 DATA 9FC1A9008D04C48D, 1003
1050 DATA 00C3A003A9208D03, 703
1060 DATA C3C8B160D0034C77, 1074
1070 DATA C1C922D00BA9014D, 894
1080 DATA 00C38D00C34C24C0, 835
1090 DATA AE00C3F0DF297FC9, 1201
1100 DATA 20B0D9B1608D01C3, 1035
1110 DATA 8C02C3AD04C4D024, 954
1120 DATA A9008D04C38D05C3, 858
1130 DATA A00238B160E9018D, 866
1140 DATA 06C3C8B160E9008D, 1048
1150 DATA 07C3A98F8D08C3A9, 1027
1160 DATA 058D04C4AC02C3A2, 877
1170 DATA 00E8C8B160CD01C3, 1106
1180 DATA F0F7E001F04B8662, 1259
1190 DATA A9008563A562C90A, 875
1200 DATA 900A38E90A8562E6, 914
1210 DATA 634C94C0AE04C4AD, 1062
1220 DATA 03C39D04C3A9209D, 912
1230 DATA 05C3A563D005A920, 878
1240 DATA 4CBDC009309D06C3, 872
1250 DATA A56209309D07C3A9, 848
1260 DATA 2A8D03C318AD04C4, 778
1270 DATA 69048D04C4888C02, 728
1280 DATA C3A000B913C2D003, 964
1290 DATA 4CEFC0CD01C3F007, 1155
1300 DATA C8C8C8C84CDBC0AD, 1460
1310 DATA 04C4C949D046AE04, 930
1320 DATA C4AD03C39D04C3A9, 1092
```

Repeated use of the same control character gives a line showing the number of characters used. For example, 29 cursor rights followed by graphics mode (CONTROL N) is shown by:

```
2039 REM   29*CUR TXT
2040 ?"[illegible]
[illegible]QWERTYUIOP
```

If there are too many characters to fit on one line, then a second line is inserted, like so:

```
2058 REM CLR-  3*CUD- 10*CUR-BLK WHT
     RED CYN PUR GRN BLU YEL ORN
2059 REM BWN RED CYN GRN BLU
2060 PRINT"[illegible]
```

Other control characters found in BASIC text are shown as 'CT?' where '?' is the ASCII equivalent of the control code. For example, sending 7 line feeds (CONTROL 'J') to a printer would be shown as:

```
3039 REM   7*CTJ
3040 PRINT#4,"[illegible]
```

The three letter abbreviations used are:

ASCII	ABREV.	FUNCTION	ASCII	ABBREV.	FUNCTION
17	CRD	CRSR DOWN	29	CRR	CRSR RIGHT
145	CRU	CRSR UP	157	CRL	CRSR LEFT
144	BLK	BLACK	5	WHT	WHITE
28	RED	RED	159	CYN	CYAN
156	PUR	PURPLE	30	GRN	GREEN
31	BLU	BLUE	158	YEL	YELLOW
18	RVS	REVERSE ON	146	OFF	REVERSE OFF
129	ORN	ORANGE	149	BWN	BROWN
150	LRD	LIGHT RED	151	GR1	GREY 1
152	GR2	GREY 2	153	LGN	LIGHT GREEN
154	LBL	LIGHT BLUE	155	GR3	GREY 3
14	TXT	TEXT MODE	142	GRF	GRAPHICS MODE
19	HOM	CRSR HOME	20	DEL	DELETE
147	CLR	CLEAR SCREEN	148	INS	INSERT
133	F1	FUNCTION KEY 1	137	F2	FUNCTION KEY 2
134	F3	FUNCTION KEY 3	138	F4	FUNCTION KEY 4
135	F5	FUNCTION KEY 5	139	F6	FUNCTION KEY 6
136	F7	FUNCTION KEY 7	140	F8	FUNCTION KEY 8

Program Listing (cont.)

```
1330 DATA 009D05C3AD04C39D, 886
1340 DATA 06C3AD05C39D07C3, 933
1350 DATA AD06C39D08C3AD07, 914
1360 DATA C39D09C3AD08C39D, 1089
1370 DATA 0AC318AD04C46907, 714
1380 DATA 8D04C438AD06C3E9, 1004
1390 DATA 018D06C3AD07C3E9, 951
1400 DATA 008D07C3AD03C3AE, 888
1410 DATA 04C49D04C3B914C2, 955
1420 DATA 9D05C3B915C29D06, 920
1430 DATA C3B916C29D07C3C9, 1156
1440 DATA 3FD008AD01C30940, 721
1450 DATA 9D07C318AD04C469, 861
1460 DATA 048D04C4A92D8D03, 703
1470 DATA C3AC02C34C29C0AE, 1047
1480 DATA 04C4F00CE88E04C4, 1026
1490 DATA A9009D03C320B7C1, 932
1500 DATA A004C8B160D0FBC8, 1296
1510 DATA 981865608560A900, 771
1520 DATA 656185614C08C020, 736
1530 DATA 33A5A52D852F8531, 788
1540 DATA A52E85308532A537, 795
1550 DATA 8533A538853460A5, 851
1560 DATA 2D8564186D04C485, 744
1570 DATA 66852DA52E856569, 830
1580 DATA 008567852EA000B1, 752
1590 DATA 64916638A566E901, 904
1600 DATA 8566A567E9008567, 972
1610 DATA 38A564E9018564A5, 953
1620 DATA 65E9008565C561D0, 1070
1630 DATA 04A564C560B0D8A0, 1114
1640 DATA 00B904C39160C8CC, 1029
1650 DATA 04C490F518A5606D, 983
1660 DATA 04C48560A5616900, 796
1670 DATA 856160114352441D, 589
1680 DATA 435252914352559D, 767
1690 DATA 43524C90424C4B05, 591
1700 DATA 3748541C5245449F, 649
1710 DATA 43594E9C5055521E, 667
1720 DATA 47524E1F424C559E, 647
1730 DATA 59454C1252565392, 649
1740 DATA 4F4646814F524E95, 736
1750 DATA 42574E964C524497, 758
1760 DATA 4752319847523299, 710
1770 DATA 4C474E9A4C424C9B, 752
1780 DATA 4752330E5458548E, 616
1790 DATA 47524613484F4D14, 490
1800 DATA 44454C93434C5294, 733
1810 DATA 494E538546312089, 655
1820 DATA 463220864633208A, 577
1830 DATA 463420874635208B, 583
1840 DATA 463620884637208C, 589
1850 DATA 4638200043543F, 372
1860 DATA END, 74057

READY.
```

The listing shows a basic loader program which reads hexadecimal statements (with checksums) and POKEs the program into safe memory at $C000 or 49152. If any error is found in the data then it is reported along with the line number in which it occurred.

Please note: save the loader program before trying to execute the machine code routine. If you wish to save the machine code program then type:

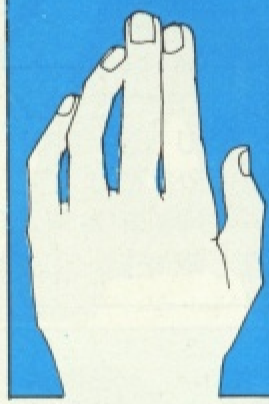
POKE43,0:POKE44,192:POKE 45,0:POKE46,195:POKE55,0: POKE56,208:C
SAVE"NAME",1
SYS 74738

To re-load the routine, type:

LOAD"NAME",1,1
NEW

To execute the routine, first load or type in your BASIC program then type 'SYS 49152' or 'SYS 12★4096' and wait a few moments (depending on the length of your program). When 'READY' appears, you can RUN, re-SAVE or LIST your new program. If the BASIC text has a line 20 containing control characters but also has a line 19, then another line 19 is inserted; the operation of the program is unaffected.

David Crisp examines Practicalc, Practifile and Inventory 64, three pieces of business software from the Computer Software Association for your Commodore 64.



BUSINESS FILE



COMPUTER SOFTWARE ASSOCIATION has been advertising these for quite a while now but this is the first opportunity I have had to try them. My first intention was to compare each program with a similar product from another company but later I decided to treat them as a complete but non-integrated package. I have been using a spreadsheet a lot recently and so **Practicalc** was the one I dealt with first.

The packing leaves a lot to be desired. It is a flimsy box which suffers *greatly* in the post. This type of packing seems to be the type with all of the programs from C.S.A. Even when stored under normal circumstances at home the box is easily crushed and soon becomes more than useless. The manual was quite comprehensive and it's small size is misleading. It is clearly set out and packed with information laid out in a clear way with a good **tutorial** approach. If you have not used a spreadsheet before then this would soon get you going. Lacking in this manual, as in most manuals that arrive with spreadsheets, is anything that would point out the potential of a spreadsheet.

For the novice it is difficult to imagine it as anything other than an elaborate calculator which can do things such as cash flow projections and analysis. When you start to plumb the depths, however, it is possible to suddenly discover how versatile the sheet can be. To date I have found no manual which stimulates the imagination. I have an unusual invoicing application on my spreadsheet which allows me to easily raise invoices and statements for account customers. I can see at a glance each account and its present position. Some crafty print routines give the appearance of a standard invoicing program. This is a function I worked out for myself while trying to justify owning a powerful spreadsheet. Enough rambling and back to **Practicalc**.

Display

Practicalc seems to load very quickly and once loaded you are asked how many columns and how many rows you would like. The default values are 40r★25c. For my own use this is not really big enough but for many small users it would be enough. After choosing the size of your sheet the top left corner of the sheet is displayed. I cannot really say the display pleased me as it looked a little bare. The columns were displayed well enough but the rows were not marked out at all, and I had some difficulty placing the cursor on the desired row. At the top of the screen where I have learned to expect information such as cells remaining and memory available there was nothing except the current row/column and the word 'PRACTICALC'.

The information there though was clear on both the 5" colour screen of an SX-64 and a 14" green screen monitor; some programs while looking O.K. on one are often almost invisible on the other.

Standard practice

As would be expected on a sheet, full mathematical functions are available for use such as LOG, EXP, ABS, SQR etc, plus a few extra ones such as AVG (for finding the mean average for a range of numbers), MAX AND MIN (for finding maximum and minimum values over a range) and COU (for finding the number of entries in a range). This is only a selection of the possibilities which are available. Once you have been introduced to a few of the functions you are led into a tutorial to show the possible use of the sheet.

Overshoot

I am a great believer in working through a manual and this one was no exception. I entered up the information as instructed and soon felt I was 'alien zapping'. The repeat rate on the cursor keys was far too fast and trying to home in on a box was as hard as getting the last alien on an invader game. It was at this point I stopped for coffee. A series of short stabs at the key soon enabled me to get around the sheet. The column numbers on Practicalc are at the bottom of the screen and on all the other sheets I have used they are to be found at the top. It took some time to get used to this but with practice I soon mastered it. Pressing function keys soon enabled me to bring up the option menu. This is fortunately fairly standard and as with most sheets a single key press enabled me to clear, resize, delete (column/row), format etc. As usual, commands could refer to the individual

cell, row, column, or to the sheet as a whole. It is possible to set the display up to show integers only, text, numerics to two decimal places and so on. To enable you to enter formulae faster there is the replicate function. This allows you to repeat a cell's formula across the sheet or down according to the range you specify. It would appear to be a vital function but one which is not present on all spreadsheets.

Pictures

A nice extra with this sheet is the ability to display sets of figures graphically. As spreadsheets tend to contain a large amount of numeric data it is nice to be able to display a whole row or column as a bar chart in order to get an overall picture. To chart a set of numbers with Practicalc was very easy indeed and quick. You have the choice of displaying in low or hi-res graphics which means a bar of asterisks in low-res or a bar of solid squares for hi-res: simple but it does the job.

Printout

It is possible to obtain hardcopy from the unit using either a Commodore printer or the built in software interface. The interface is mentioned in the manual but there is no **specification**. If the specification was there it should be possible to make any printer work using a custom built cable. I could get some of my leads to work and not others, so all I can suggest is to try before you buy.

Practicalc conclusion

I can't quite make up my mind on this one. It is definitely not a **bad** spreadsheet but on the other hand it is not brilliant. If you are a first time user and were learning to use one I think you would take to it quite easily. However, for someone who is used to a spreadsheet I think they may have a few difficulties in getting used to some of the odd quirks, such as column identifiers being at the bottom. I am suffering from a severe bout of indecision. The best thing would be to have a look at it and decide once you have seen it. I would put it somewhere between Busicalc and Richard Shepherd's Autocalc.

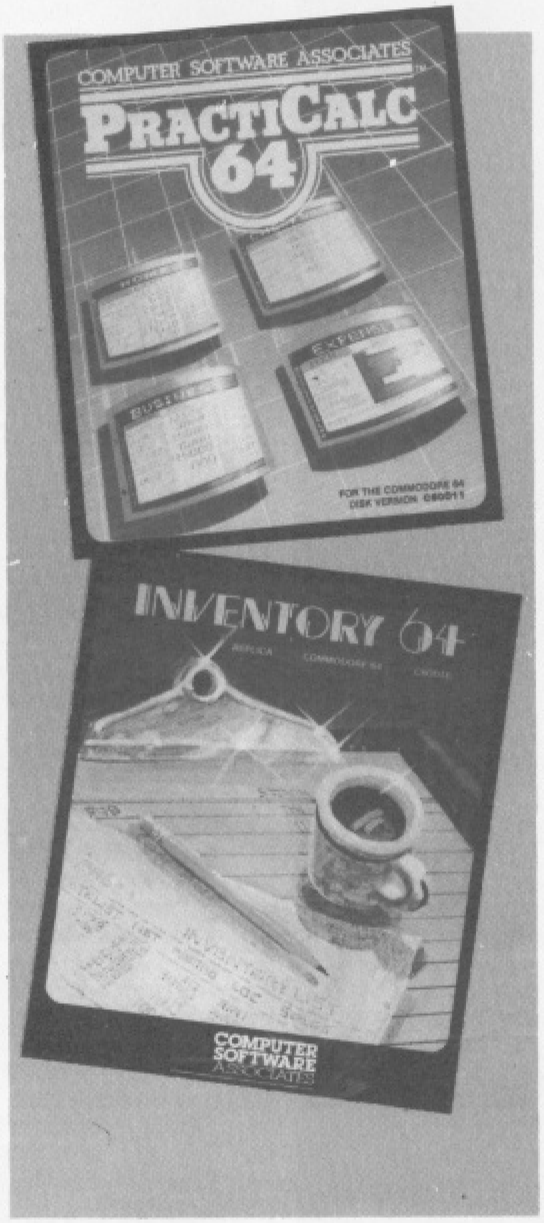
Inventory 64

From the same company comes INVENTORY 64. My comments about packaging are the same as with Practicalc. The manual is considerably smaller but the program is considerably simpler as well. It is an inventory control system in the words of C.S.A. and is straightforward to use even without the manual. It is menu driven and getting used to it only takes a short while. It is what I call a preformatted database and so is inflexible but that is all some users will want.

Capacity

You are able to hold approximately 650 items per disc but there is no reason why more than one disc could not be used. I will deal with the program in the order in which it is discussed in the manual. After loading, the first thing that is required is that you insert the data disc. Relative filing has been employed in this program and so the setting up of your first disc will take some time although, once in use, searching for specific records is fast.

The first menu to appear on screen gives you the option to exit, zero sales to date, maintain or produce reports. Choosing option one (maintain) gives you the opportunity to manipulate the records. When entering stock for the first time you will come across the first restriction. It is necessary to allocate the stock item a number between 1 and 650; this cannot be prefixed or suffixed with a letter or anything else and so a stock identifier will be a meaningless number. This is laziness from a programming point of view. It is possible to use a hashing alogrithm to produce the record number and also to find it again and the algorithm can be a simple formula. Any program I have written using relative files has been written using a hashing algorithm and so identifiers can be precise or at least meaningful. Another problem caused by the lack of the algorithm is that you are unable to use the numbers from 337 through to 358. This again is indicative of lazy programming, but could be put right in later versions.



Standard options

All the functions you would expect of a simple stock control program are there. You can produce a re-order list, on predetermined levels, a price list of all stock held; a stock list for reference etc. As I said before, it is a simple program and on the whole performs well. A nice touch is being able to enter the unit of measurement so it is possible to indicate whether something is stocked by size, weight, quantity, and so on. On this one it is possible to zero all sales to date, an important feature which has been missed on a couple of other stock control packages I have looked at.

Printer compatability

This program is written in BASIC. For a program of this type, BASIC is perfectly adequate. The other advantage is that because machine code routines are not stashed away in corners of your RAM any printer interface you may have will probably work. My interface software sits at C000 (hex) and so worked OK. Printouts were simple and clean with everything clearly marked. Total parts in stock and total stock value was clearly marked and everything relevant was printed.

Inventory 64 conclusion

A simple program which worked well in what it covers. It is lacking in a few things, e.g. the use of stock numbers and it was not possible to perform any batch operations so if the VAT rate changed and your price included VAT it would be necessary to change each record. It is a stand alone system and so you must assess whether it will save a lot of time. I feel that it lacked a daily sales printout which would have made it useful in applications where a storeman could use it for issuing parts within the factory etc. With the addition of these things I feel it would be a very useful program. If you were to consider buying it then I recommend that you write down the information you would want from an inventory control package and then see if this program provides what you need. All in all it's not bad for the price.

Practifile

From the advertising information Practifile was the package I was most looking forward to using. I am a fan of database management systems and I spend a lot of time writing odd applications on them. They tend to be extremely versatile and can usually be used in a simple BASIC menu driven form or alternatively programmed to perform many operations automatically.

As the manual is fairly thick it represented a couple of hours bedtime reading before actually loading. I would not recommend reading the manual straight through unless you are conversant with this type of program as they can be heavy going and you may end up quite lost, but as so many database

management programs are similar I tend to read through the manual first in order to see if there are any extra tit-bits that have not been present on other systems.

The packing and the manual, unfortunately, are much the same as with the other two programs. The only difference seems to be the thickness and the number of errors. My copy came with a six page addendum, not to correct mistakes but to clear up parts of the manual and to explain possibilities not explained fully in the manual. Most of the pages are concerned with the tranportation of files and data from Practicalc into Practifile and vice-versa. This indicates that some degree of integration is possible between the two programs and other sequential files. Full marks for explaining it so well, but could it not have been concluded in the manual as loose pages are so easily lost.



In use

This is only a review and not a full length depth trial I will only point out the main features of this package. If you would like to find out more about what these packages are capable of and how they can be used then it may be worth reading a couple of the many good books which are available on database management systems.

After loading, the first thing I noticed was the constant need to change discs. From reading the manual I realised there would be a need to change discs but did not fully appreciate just how often the need would arise.

As a simple test I decided to set up a fairly basic file which would enable me to store my ever increasing stock of software. I wanted some degree of mathematical calculation regarding total values etc and the ability to search according to program type. Practifile does all this as I expected it would but to be honest the way it did it was dismal. The screen displays and prompts were tatty and simply seemed to scroll up as you went further through the format routines. I found myself having to draw what I wanted on paper and then translate this into a form that Practifile could understand. I seemed to be relying a lot on memory regarding what I had already put in and if I had made a mistake it required going through each field defined and confirming that each one was alright.

Menus were comprehensive but their layout was basic to say the least and throughout using it I constantly had the feeling that I was testing a program to see if it was functioning OK before sending it back to be tidied up and presented properly. Unfinished is probably the best way of describing my feelings about this program.

Plenty of space

One thing that cannot be criticised is miserly use of filespace. It's capacity seemed endless with an advertised number of 3800 files per record. Obviously a record has to be of a reasonable size in order for that many records to be stored but by anyone's standards some impressive crunching has been employed.

You have a choice of filing methods (sequential or relative) and depending on the type of file you desire the speed of operation will change: a nice feature which could be employed on other databases. Relative files are fast but if the records you have are small enough to fit into RAM then obviously searching and sorting etc are much faster. Despite my criticisms, its design features such as this are impressive. That is one of those small things that make a complete read of the manual worthwhile.

It is possible to generate virtually any type of report from your records and this is fairly easy to do.

Printing

There is a section concerned with customising Practifile which resaves certain information back to the program disc; these parameters are used each time you use PRACTIFILE. It covers screen paper and ink colours, whether you are using two single drives or dual drives etc. and the type of printer you are using. As before, I found that some interfaces would work and others would not and so I am afraid I can only (once again) suggest that you try before you buy.

Another feature of Practifile is the facility for using a cassette. I could not try this as I use an SX-64 but, for long term archival storage of sequential data, tape is ideal as it is fairly safe and very, very cheap. Again this is one of those little things that are changing my opinion of Practifile.

Yet another of those important but often neglected little things is the ability to batch file. If you are entering a lot of data even random filing seems to take a long time and you are soon conscious of the delay between finishing an entry and being able to start another one. Once again, Practifile saves the day. If batch entry is chosen, the data that you type in is temporarily stored in RAM and, at the end of entry, it is organised and stored into a random file. It is surprising just how much time this can save.

Practifile conclusion

I am still a little numbed at the number of times I needed to change discs and totally anaesthetised by the 'grotty' screen display but, before all you Practifile devotees have me hanged, drawn and quartered, may I add that I think this program is very good: the things I have covered in 'bits and bobs' make all the difference. Although I will continue to feel that this program could be greatly improved, it is, without any doubt, a workhorse. I grew to like it and will continue to use it for some applications.


DEATH STAR
INTERCEPTOR
Earth is threatened by an Empire Deathstar, can you penetrate its defences and destroy it before it destroys you?
52K of pure machine code to give you 12 screens of 3D graphics, super smooth 3D animation, unbelievable sound effects, music, 4 skill levels and hiscore table . . . this is the ultimate challenge!
This game has to be played to be believed! You have not seen what the Commodore 64 is capable of until you have played Deathstar Interceptor! - Commodore 64
Available now at £9.99!
Graphically Amazing!
NOW on NOVA LOAD SUPERFAST LOADING SYSTEM
SPECTRUM VERSION AVAILABLE SOON!
DEATH STAR INTERCEPTOR
System 3 Software
Graphically Amazing!
DO YOU WRITE YOUR OWN PROGRAMMES?
If you have a marketable programme send it to us for evaluation. Excellent Payment Terms and Marketing plans. Send to Product Development Manager at the address below.
● DEALER INQUIRIES
(01) 587 0873
● EXPORT INQUIRIES WELCOME
available at Boots, W. H. Smiths, Rumbelows, Curry's and all good stores.
HOW TO ORDER
To purchase Deathstar Interceptor, simply fill in your name and address on a piece of paper, enclosing your cheque made payable to: SYSTEM 3 SOFTWARE, and post to the address below. Please allow 7 to 14 days for delivery.
Overseas orders
Please add £1.00 per game ordered.
System 3 Software
South Bank House, Black Prince Road, London SE11
Tel: (01) 587 0873

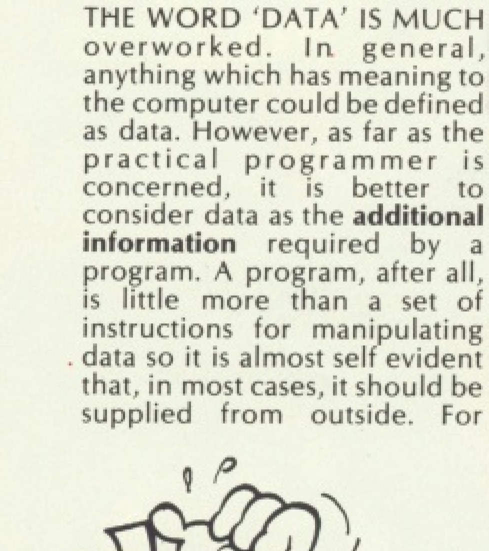

MIND GAMES
AMERICAN FOOTBALL
A fast & furious graphics strategic simulation
£9.99
FREE RULE BOOK
FOR THE SPECTRUM 48K, CBM 64
OTHER GREAT MIND GAMES:
MIND GAMES QUEST FOR ETERNITY
Starring The Overlords of the Universe
The candidate (you) have to get to the Chamber of Creation. It's a laugh a minute, since it's 2,000 light years away on the most horrible planet in the Universe ... and your starship doesn't work either!
MIND GAMES ORION QUEST
A full feature adventure starring well known nasty aliens the Zarps. Can you play the hero and stop their plans to blow up the earth.
MIND GAMES STAR FORCE SEVEN
Starring The Zurgs
After a desperate space battle only one fleet of heroes remain to prevent the invasion of earth. The future of humanity lies with you!
Argus Press Software Group
For mail order, write with cheque/P.O./card No. to:
Mind Games, Argus Press Software Group, No.1 Golden Square, London W1A 3AB.

THE BASIC FACTS PT.4



Your Commodore's series on BASIC continues with A.P. and D.J. Stephenson studying data and arrays.

THE WORD 'DATA' IS MUCH overworked. In general, anything which has meaning to the computer could be defined as data. However, as far as the practical programmer is concerned, it is better to consider data as the **additional information** required by a program. A program, after all, is little more than a set of instructions for manipulating data so it is almost self evident that, in most cases, it should be supplied from outside. For



example, a program which added the numbers 34, 568, 243 and 123.45 together and printed out the result may be of slight interest during the first run but subsequent runs would provide no further information. On the other hand, if the numbers were supplied from outside, so that subsequent runs acted on different numbers, the program would, in spite of its simplicity, have some justification for being written. As a further example, it would be a pathetic waste of time to write a program for calculating the number of factors in 456. Far better if the program could find the factors of any number supplied from outside during a run. Now we have defined data as information required by a program, it is important to be aware of the different ways in which it can be supplied. We can use the INPUT, INPUT#, GET, GET# or READ/DATA statements.

The INPUT statement

When a program is running, the instructions are executed one after the other at lightning speed — far faster than humans can think or act. If a program requires data during a run, there must be some provision for halting the computer, until such time as the user of the program has finished entering the required data via the keyboard. The relevant BASIC statement which halts the computer until you feed in some data is INPUT. There are actually two forms of this statement, depending on whether a **prompt** message is to be displayed on the screen or not. The syntax of the simpler form is as follows:

```
INPUT variable
```

INPUT is the BASIC keyword and must be followed by a variable name of your own choice. For example, suppose we write the line:

```
100 INPUT A
```

The computer will halt at line 100 and outputs the character '?' to the screen accompanied by the winking cursor. When you have entered suitable data and pressed RETURN, the data will be placed in the variable named A and the computer will resume its high speed progress. If the data is unsuitable, an appropriate error message is displayed. The commonest cause of error is when string data is entered instead of a number, in which case the somewhat cryptic error message 'REDO FROM START' will appear on the screen. If strings are to be entered, the variable must be a string variable such as INPUT A$. It is worth remembering here that numbers are accepted into a string variable (because numbers are included in the definition of strings), numeric variables will accept no other characters but numbers. If the operator happens to press RETURN before entering data, any original value which the variable may have had is still preserved. It is allowable to enter many data items at a time into different variables, providing they are all separated by a comma. For example:

```
100 INPUT A,DF
```

You enter the A data first, followed by RETURN, then the



DF data, followed by RETURN. Although this facility is offered, you are advised not to use it very often because it could be confusing to the operator. It is far better to use a seperate INPUT line for each data item. In fact, we should mention that the use of the simple input statement described above should be employed sparingly. A halted program, accompanied only by insolent looking '?' can strike terror into the heart of an inexperienced operator — remember the person using the program may not be computer minded. What is needed is some message to accompany the INPUT statement which guides the poor user. In other words, he or she must be prompted. Because of this, it should be almost a rule that you use the following expanded format of the INPUT statement:

```
100 INPUT"message";
variable
```

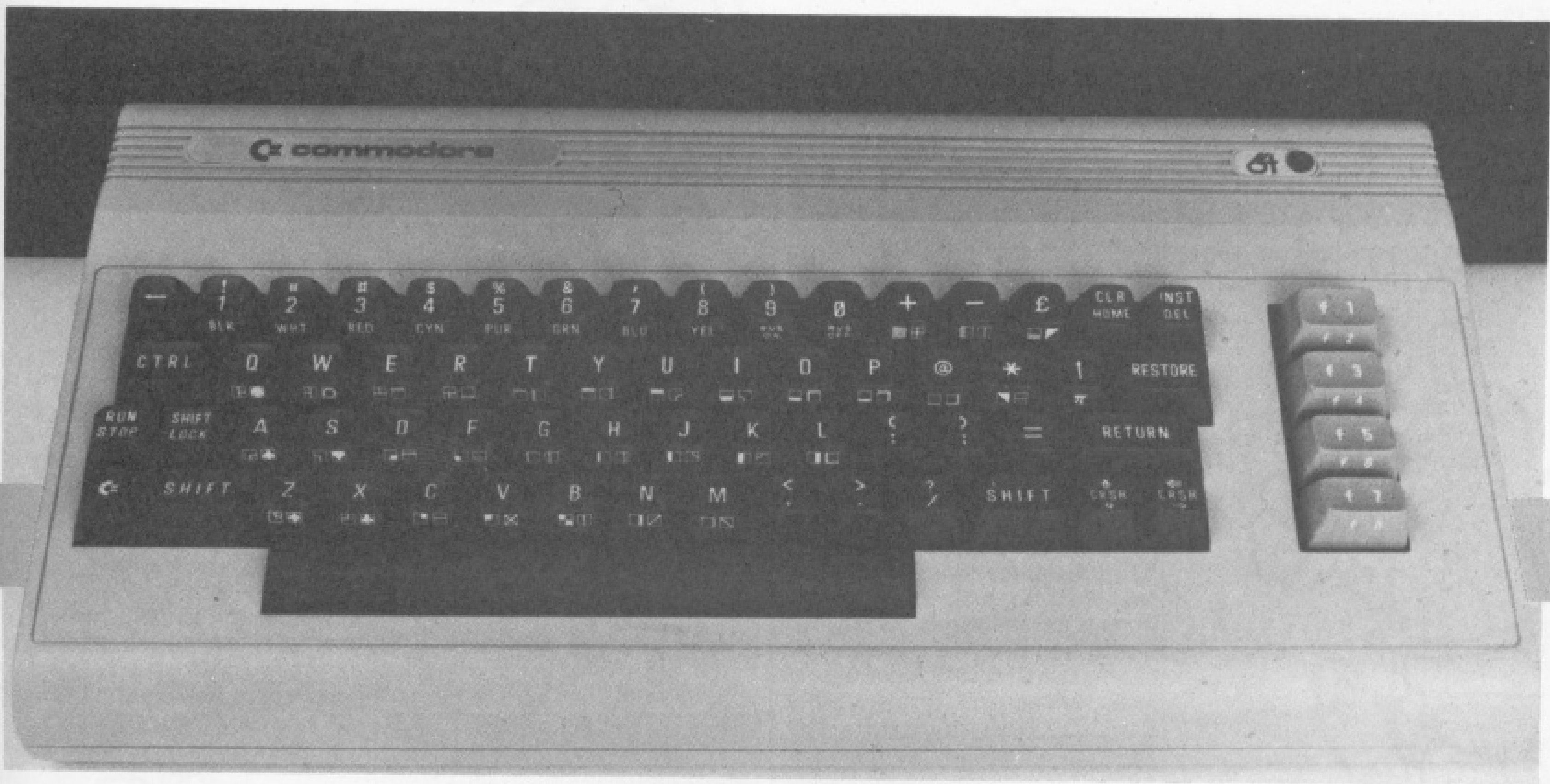
The computer will now halt as before but this time, a suitable prompt message is displayed for guiding the operator. For example:

```
100 INPUT"Enter gun
velocity ";GV
```

Note that the message must be enclosed within double quotes and that a semicolon must separate the message from the variable. Notice also that a space has been left between the end of the message and the closing quotes. This is not mandatory but, if you leave it out, the data entered will butt up close to the end of the message and look messy on the screen.

INPUT validation

Sometimes, due to carelessness, malice aforethought or sheer ignorance of the



consequences, operators will respond to an INPUT statement by entering data which, although correct in syntax, is quite daft. For example, if the prompt is "Enter your age" and the operator responds with 456 then the accuracy of the data is, to say the least, suspicious. The computer will accept it of course, because it is naive and quite incapable of mature judgement. It is up to the programmer to include validation checks on all data received from the keyboard.



The best place for this is immediately, or at least soon, after the INPUT statement. We will take the age query as an example of a validation check:

```
500 INPUT"ENTER YOUR AGE ";AG
510 IF AG> 120 OR AG <2 THEN GOTO 500
```

Line 510 would reject ages greater than 130 or less than 2. These limits are naturally up to the programmer to decide but it is hardly likely that a child less than two years old, however precocious, would be operating a keyboard. At the other end of the scale, it is difficult to imagine how someone over the age of 130 would have sufficient strength left to push down the RETURN key. However, it is often desirable for the program to include an escape clause. For example, it may be that the operator may want to see what happens if he was 131 years old — a kind of digital peep into the future. To cater for such awkward individuals, the previous example could be modified as follows:

```
500 INPUT"ENTER YOUR AGE ";AG
510 IF AG>=2 AND AG<=130 THEN GOTO 540
520 INPUT"ARE YOU SURE Y/N) ";K$
530 IF K$<>"Y" GOTO 500
540 ...rest of program...
```

This version incorporates the 'Are you sure?" trick. That is to say, if the data supplied is outside the limits, it is finally



rejected only after the operator has been given the chance to confirm or reject it by entering either Y or N (yes or no). Notice that line 510 uses the AND connective in between the two limits instead of the OR. If you think this is wrong try using OR instead and see what happens. You can get into some strange difficulties with AND and OR. For example, if a number is less than 130 OR greater than 2 then it could be any number within the range -infinity to +infinity. This would be a ridiculous pair of limits, but if we had used AND instead of OR, the number is tightly bound. We shall be dealing with the AND, OR and NOT connectives in great detail later in the series when we discuss logical operations. In the meantime, see if you can re-arrange the program in a better way. Remember though, if you have to use more GOTOs than we have done it should not be considered an improvement.

The INPUT statement

The normal INPUT statement is used for entering data from devices other than the keyboard. Data can also be stored on a data tape or tape disc and read in from the program by means of the INPUT statement. However, we do not think this is the best place to treat the subject because it demands a knowledge of channel numbers and other unpleasantries. INPUT is best treated later in the series under the heading of tape and disc files.

The GET statement

The INPUT statement is used for entering one or more characters from the keyboard, the RETURN key being used to inform the computer that no more are to follow. The GET statement is different. It is used to enter a **single character** and does not require the RETURN key. As soon as we press a key, the character is entered immediately into the variable. Thus, if we write GET A or GET A$, any key pressed is entered immediately into the variable A or A$. However, some strange things can happen unless you understand the action of the **keyboard buffer**. This is a small memory, capable of holding only ten characters. Whenever a key is pressed, the character code of the key is entered into the buffer. When we use the

GET statement, one of the characters is read into the variable, leaving room for another one to be entered into the buffer. If more than ten keys have been struck without an intervening GET, the excess key presses are lost. The keyboard buffer action still goes on, irrespective of other tasks the computer may be



performing at the moment. For example, if a program is running and we idly press a key, nothing may appear to happen but the character will have entered the buffer (providing it was not already full up). If, later on in the program, a GET statement is reached, the character will be scooped up from the buffer without you knowing it. If we wish to use GET in order to pick up a particular character, we must include a small loop which halts the computer until the required key is pressed. For example:

```
500 GET K$: IF K$<>"F"
THEN GOTO 500
```

This is an example of a line which GOes TO itself unless some criteria is satisfied. The only way for the computer to escape from the loop is by pressing the 'F' key. We can widen the criteria to include any key by using:

```
500 GET K$: IF K$<>"" THEN
GOTO 500
```

Note carefully that "", which is called the **null string** (no key at all), must not have a space between the quote marks. The computer stays in the loop until the keyboard buffer receives a character — it doesn't matter which character. If we wanted the computer to halt until the space bar was pressed, then we would leave a space between the quotes.

The GET statament

This is used when we wish to read a single character from a device other than the keyboard, such as from tape or disc. Like INPUT #, we shall postpone discussion on GET # until later in this series.

The READ/DATA statements

We have defined data as information required by a program and, so far, we have assumed that such data has been received from the keyboard during a program run. The READ/DATA statements represent a grey area because, although they are used to pick up data, the data is part of the program — it **occupies program lines.** The format of thw DATA statement is as follows:

```
DATA list of constants
separated by commas.
```

For example:

```
5000 DATA 30,50,0.5
```

Note they are constants because they have not yet been allocated variable names. Note also the high line number, which is of course an arbitrary



choice. Although DATA statements can be placed anywhere in the program, it is customary, indeed desirable in the interests of structure, to place DATA statements at the end of a program so they stand out and can easily be changed. To read the data into variables, corresponding READ statements must appear, the format being:

```
READ variables
```

For example, to read in the



three constants used in the previous DATA example, the following line must appear somewhere in the program:

```
200 READ A,B,C
```

This will read the DATA items into the three variables in **strict order**. That is to say, 30 will go into A, 50 into B and 0.5 into C. There must not be more variables in the READ than there are constants in the DATA. For example, if we had written READ A,B,C,D the computer would get cross and spit out the error message 'OUT OF DATA'. Another way of provoking an outburst is to attempt to READ the same data twice, It is not necessary to read all the data items at one go. We could have written our example in this form,

```
500 READ A
540 READ B,C
```

Although, we have used numeric constants to illustrate the features of READ/DATA, there are no restrictions as to the kind of constant. They can be simple string variables or even long sentences and can be mixed with numbers. However, there are two provisos. Any strings in DATA statements must be enclosed within double quotes and the corresponding READ variables must be string forms. A possible DATA line could be:

```
1000 DATA "CALDY",
"GREASBY","HESWALL",
"BURTON ON TRENT",540
```

This could be read with:

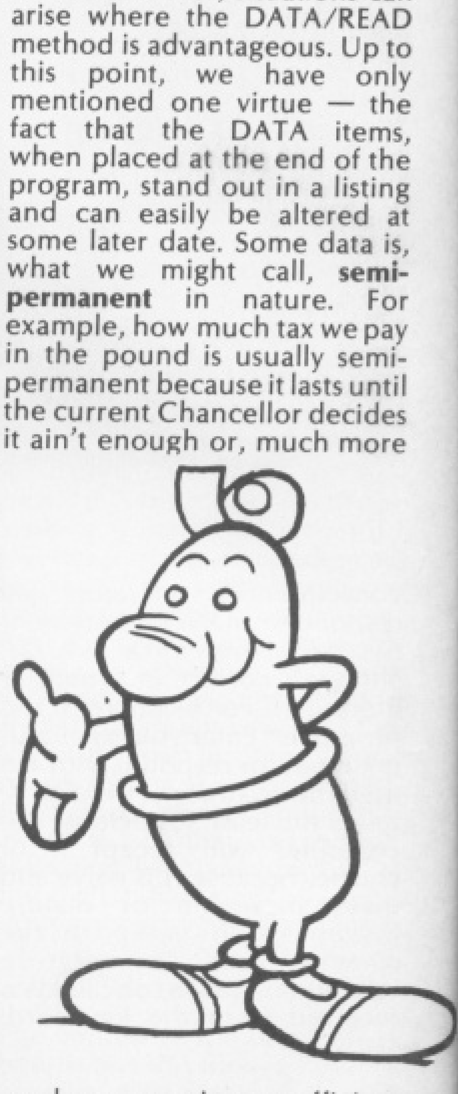
```
500 READ A$,B$,C$,D$,G
```

Note the first four are strings and the fifth numeric.

It may be asked why we could not have written the whole thing much more simply by using a set of assignments such as:

```
500 A$="CALDY":B$=
"GREASBY":C$="HES
WALL":D$="BURTON ON
TRENT":G=540
```

The answer is that, in this simple case, we could have done. However, situations can arise where the DATA/READ method is advantageous. Up to this point, we have only mentioned one virtue — the fact that the DATA items, when placed at the end of the program, stand out in a listing and can easily be altered at some later date. Some data is, what we might call, **semi-permanent** in nature. For example, how much tax we pay in the pound is usually semi-permanent because it lasts until the current Chancellor decides it ain't enough or, much more



rarely, we are given a sufficient rise in salary to lift us into the next tax bracket. If the tax percentage appears as a DATA item, it can readily be changed without scanning through the listing to find an obscure variable assignment. However, we must admit that these points alone are not sufficient to justify the READ/DATA method. To appreciate its other uses, we must delve a little deeper into other ways of holding variables.

Variable arrays

Up to now, we have only dealt with simple variables like A or A$ because they are relatively easy to understand. The time has come for us to delve into variable arrays. You will find these provide a far more flexible and powerful way of

expressing variables and well worth the extra effort required to understand them. To start with, let's see what a simple array variable such as A(n), or A$(n) if it is to hold strings, looks like. The general form is:

```
Array name(numeric
subscript)
```

For example, we can store something in A(1) and something else in A(2) etc etc. In fact we can store something in A(200) if we wish. The array name is, in this case, 'A' which may be considered as a blanket



name for a block of separate data. Each individual item in the array is identified by a number, known as the **subscript**. We must be careful to distinguish between A2 (which is a simple variable) and A(2) which is the 2nd variable in an array called 'A'. It is the presence of the brackets around the subscript which informs the computer that it is one **element** in an array. The subscript can be any number from 0 to 32767 but, unless specifically wanted, it is less confusing if you avoid using the zero subscript. Most people, except computer fanatics, like to count from 1 to n rather than from 0 to n-1. To get used to the feel of variable arrays and to consolidate previous discussions on DATA/READ statements, study the following few lines which will read data items into the string array A$(n):

```
500 READ A$(1):READ A$(2):READ A$(3):READ A$($)
1000 DATA LONDON, COPENHAGEN, BERLIN, MOSCOW
```

Type this in and run it. Nothing appears to happen because all we have done is to place the data items into an array. However, it is easy to confirm that something has happened if we now use PRINT A$(2) in direct mode — it should print our "COPENHAGEN". If there is a long list of DATA items it would be more professional and certainly quicker to use a simple loop to read in the data. For example:

```
500 FOR N = 1 TO 9
510 READ A(N)
520 NEXT
1000 DATA 3,6,13,8,20,7
1100 DATA 450,200,50
```

When this is run, the first revolution of the loop places 3 in A(1), the next revolution places 6 in A(2) and so on. To prove it, try a PRINT A(9) — you should see 50 displayed on the screen. You should try to get into the habit of using PRINT in direct mode to test what happens after a run because it can become a powerful debugging weapon. If you think that some variable should have a certain value, a quick PRINT of the variable will confirm or reject your beliefs. After all, there can be a difference between what you think should be in, say, A(4) and what actually is. If you want to hurl obscenities at the computer by all means do so but, in the end, you will find it is something you have overlooked.

The DIM statement

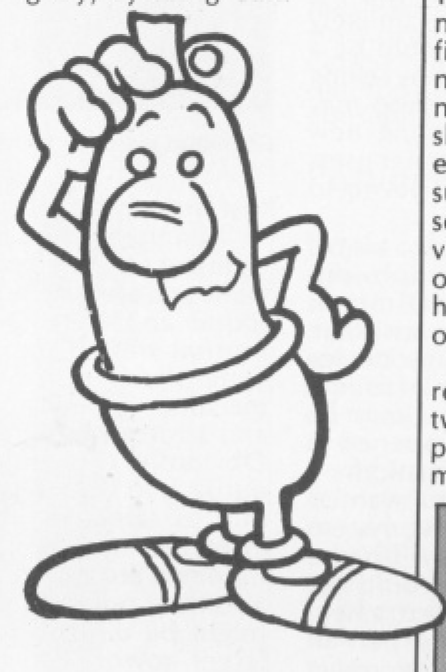
In the last example, notice that the loop was from 1 to 9. The loop was deliberately kept to a modest size to avoid having to **dimension** the array. It is a peculiarity of any array that the computer must be prior informed if the number of elements in the array is to exceed eleven, that is to say, the highest allowable subscript is 10 (this allows for the 0



subscript). The computer is informed, preferably somewhere near the top of the program, by the DIM statement, the general format of which is as follows:

```
DIM array name (highest
subscript)
```

For example:
DIM A$(100) will reserve 101 locations to hold the variables with subscripts within the range 0 to 100. Once an array has been dimensioned, you must now allow the program to re-dimension or you will get the error message, REDIM'D ARRAY. This is understandable when you think about it. Once the computer has gone to the trouble of allocating its resources for your array, it is going to be very annoyed if you later change your mind within the same program. If you must re-dimension, you must first clear the computer of all variables (quite a far reaching excercise and not to be taken lightly) by using CLR.



Multi-dimensional arrays

The type of array variable we have discussed so far is classified as a one-dimension array. It is possible to extend this idea to include arrays of two or more dimensions. A two dimensional array has the general format:

```
Array name (1st subscript,
2nd subscript)
```

For example, A(3,4) is one particular variable in the two dimensional array named 'A'. It is a little difficult to grasp the computer's concept of two dimensions but suppose we consider a matrix of numbers as follows:

3	5	2	4
6	1	0	4
7	9	8	1

It consists of three **rows** of four **columns** and could be described as a 3 x 4 matrix. We could then imagine this stored



in an array A(rows,columns). Thus, A(2,1) would hold the number 6 because it is in the first column of row 2. The number 8 is in A(3,3) and the number 5 is in A(1,2). We should stress that it is not essential to allocate the first subscript to rows and the second to columns. How you visualise or use the array is your own choice — the computer hasn't got a clue what a column or row is anyway.

The following few lines will read in the 12 data items into a two dimensional array and prints them to the screen as a matrix of 3 rows and 4 columns:

```
500 DIM A(3,4)
510 FOR R = 1 TO 3
520 FOR C = 1 TO 4
530 READ A(R,C)
540 PRINT A(R,C);
550 NEXT
560 PRINT
570 NEXT
1000 DATA 3,5,2,4,6,1,0,4
1010 DATA 7,9,8,1
```

The inner loop, lines 520 to 550 begins with R fixed at 1 while the first four constants are read in and printed. The next four constants are similarly read and printed but this time with an R value of 2. The last four items are read and printed with an R value of 3. Note the semicolon, which terminates line 540, ensures that each row of four items is printed on the same line. The separate PRINT statement at line 560 is to cancel the semicolon effect in order that the next row starts on a new line.

COMPUTERS IN BUSINESS

In most small towns, computers usually have to share shop space with an assortment of other electrical goods. David Crisp visited Minehead Radio where this is the case.

IF YOU LIVE OUTSIDE A LARGE TOWN, the chances are that you do not have a shop close-by which specialises solely in home micros. However, you can probably buy micros, peripherals and software from your nearest dealer who will be either one of the national chainstores or a shop which sells computer related products as an extension to its existing business, normally, an audio/video/electrical shop.

Trained staff

In many towns and villages, it is unlikely that a 'computer-only' shop would be a viable economic proposition so, by selling computers as a 'sideline', the shop may increase its turnover by attracting new customers. It also enables computer users to purchase equipment without having to travel long distances.

Despite this being a godsend to a lot of people, this system entails obvious disadvantages. The most common problem stems from asking an assistant, who may be as much a computer novice as yourself, for advice on buying or using a computer. In some cases, you may be given incorrect or, as once happened in my case, very dangerous information.

In my early computing days, I wanted to control my domestic lighting system with the computer: fortunately, I had a good idea of what I could and could not do but, when I asked for a little extra help at a shop that professed to be 'expert' in these matters, I was advised to connect the user port direct to the 240v mains. Needless to say, had I followed their advice, I would not be here to tell the story.

Staff training in shops is gradually improving. The large chainstores, on the whole, provide basic but important training in computer sales and use. Gone are the days when, if you asked about 'resolution' the staff wondered why you were inquiring about their plans for the new year!

Minehead Radio

One of the many small shops which has taken home micros under its wing is Minehead Radio. It is situated on the borders of Exmoor and serves the people within a radius of about twenty miles. In the summer, its business is boosted by a nearby holiday camp but, on the computer front, the locals provide the vast majority of its business.



Graham Lawrence of Minehead Radio

Minehead Radio received its first computer ten months ago — a Commodore 64. Suppliers were rapidly found and, very soon, the whole back portion of the shop was taken over by computers. It stocks a wide range of machines including the 64 and the VIC 20 and a corresponding amount of software. Obviously, it is unwise for a shop in the outback to stock too many copies of each title as sales per copy are fairly small. Printers, a wide range of floppy discs and monitors are also stocked by the shop; it is also their policy to stock many items that might be difficult to obtain outside the larger towns. For example, how many shops do you know where Commodore printer interfaces may be bought off the shelf — certainly not many in Somerset?

A large comprehensive school lies within walking distance of the school. Come 4.00 o'clock, it is difficult to get near the shop as children file through its door in their eagerness to try out the latest game.

Minehead Radio is usually successful in sorting out its customers' problems. Graham Lawrence, the financial director of the company is becoming more conversant with each computer. He says "It is easy to get familiar with our other lines such as videos, T.V.s etc but, as each computer comes out or, as new peripherals hit the market, it is easy to find you are devoting all your time just to keeping up with what is available and that's without the constant release of new software".

Another problem confronting small retailers is the relatively short shelf life of many of the games. Graham states: "When we hear that a new release is available, it is very difficult to assess the number of units to stock as it is easy to find you have over ordered and, just a few weeks later, you are left with a box full of 'dead' games. Apart from that it is obvious that a lot of copying goes on and this has an obvious effect on the number of sales per game. Copying is a problem and the people who are copying are only doing themselves out in the long run. If a simple way of making programs 'copyproof' could be found and our sales increased, we would obviously be able to take on a wider range of software".

From the conversations I overheard when I was in the shop, it is obvious that copying amongst friends is a common practice.

All muck in

Nobody in Minehead Radio deals exclusively with the computer department. Andrew Jordan and Keith Middleton, who are usually found in the van delivering T.V.s, videos and microwaves, often have to turn their hands to loading and demonstrating the latest releases. While they haven't got the time to become experts in this job, Andy and Keith have had Commodore 64s at home in order to learn the basics.

Kate Broderick is responsible for looking after the main computer system in the back room. She too shows a keen interest in home micros but, again, she faces the problem of not having enough time to keep up with the home computer department.

"Everybody in the shop has to muck in when things get hectic", says Graham Sizer, the director. "It is obviously not the best arrangement but we can usually sort out most problems that arise".

Hard to get

From talking to the two Grahams, it is obvious that difficulties arise in the shop due to the unreliability of supplies. "Take last Christmas", remarks Graham Sizer, "Commodore 64s were available in abundance, unlike other machines, but it was nearly impossible to get a cassette unit. We were stuck with unsaleable computers. It was possible to get the 1541 drives but how many parents could afford to buy their children a 64 and a disc drive in one go?



Pete, who works in the service department makes similar comments about the components: "I could service most of the machines we have here but the supply of I.C.s is so irregular it is very, very difficult to maintain an adequate stock".

The home computer industry is relatively new and still sorting itself out but, what annoys the two Grahams more than anything is the way that manufacturers and distributors seem to 'dump' the small retail outlet in favour of the large multiples when things get short or at peek times like Christmas. "There have been times when we have not been able to get any computers at all. It gives the impression that we are not doing our job properly".

Up to the ceiling

At the rear of the shop, where the computers live, it was easy to see that space is a problem. The software was displayed from floor to ceiling and, when the shops gets very busy, it is easy to lose a considerable amount of stock through shoplifting.

"Not any more", Graham Sizer tells me, "We did notice that quite a lot of stock was disappearing but we now have closed circuit T.V. ...It does have a deterrent effect as losses have dropped considerably since it was installed".

Putting the 64 to use

Down at Minehead Radio's video library, a few hundred yards away from the main shop, Ron Fitzer is concealed under a pile of video tapes. Ron says: "We have got about 1500 tapes at the moment and that number is increasing all the time. We do need to keep some type of inventory in order to get stock values etc so we are going to use a Commodore 64. I have one at home and use it a lot and I know that it is capable of doing all that I require."

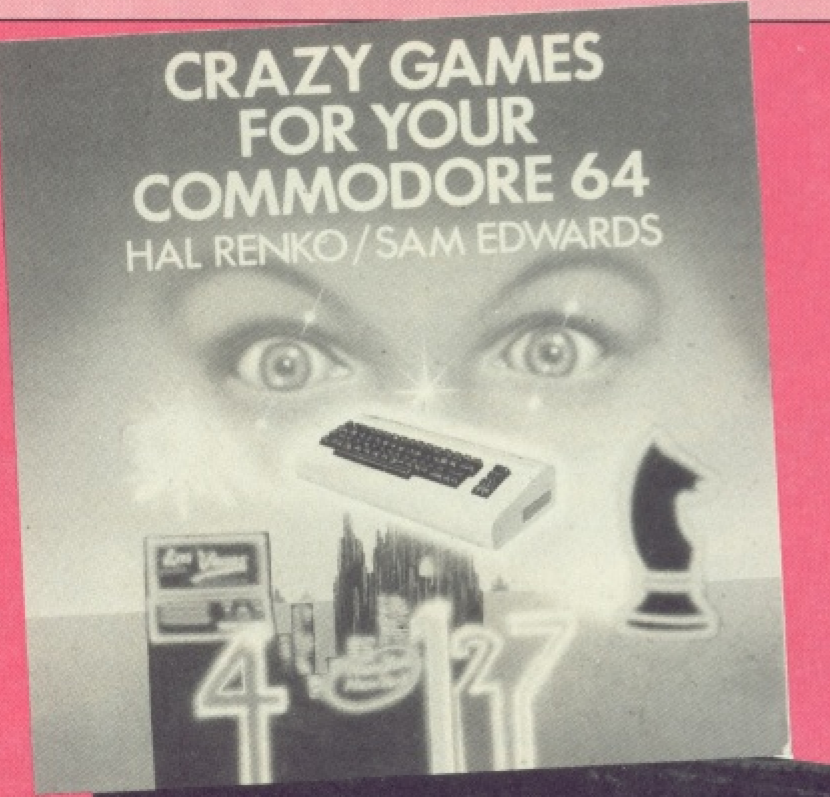
While talking to Ron, it also became apparent that using the 64 and a piece of software such as Superbase, he should be able to keep an eye on which titles and types are most popular. With a good member list and a video list, it should also be easier to keep members informed.

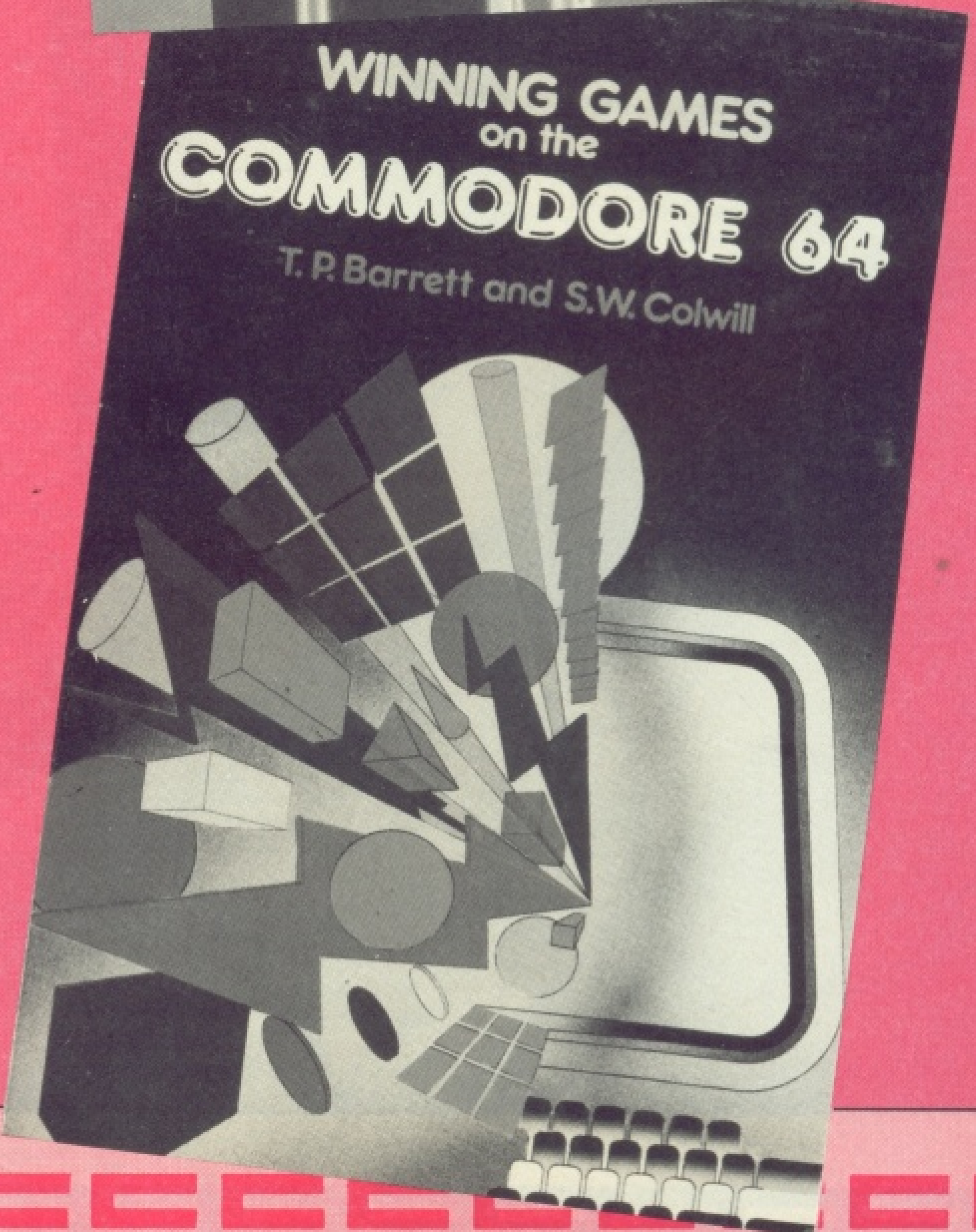
Minehead Radio seems to be investing more and more in computers. They are obviously providing a service that is required in the area and seem to be doing it with a high degree of success. It was encouraging to see the 64 being used so much by the shop's staff who have access to a wide range of machines: Ron, for example, as secretary of the local rugby club, is storing the club's records on his own Commodore 64. This only goes to show what a versatile machine the 64 is. Graham is also keen to start stocking the new Commodore models.

For a small shop, Minehead Radio offers a wide range of services. Retailing home micros is obviously a much harder task than would at first appear. It is time consuming and, so far as capital investment goes, expensive. I finally left at 4.00 o'clock to be replaced by the local comprehensive's computing contingent.

This month we show that reading can be all fun and games as Alan Webb assesses some of the Commodore games books on the market.

REFERENCE LIBRARY





THE MAJOR PROBLEM ENCOUNTERED by new-comers to computing when they first get their computer is . . . "OK, what do I do with it". Quite understandably a ready written game or other program to type in is what they really need. In response to this requirement, there has been a veritable flood of books providing games listings onto the market. Here is a small selection of such publications.

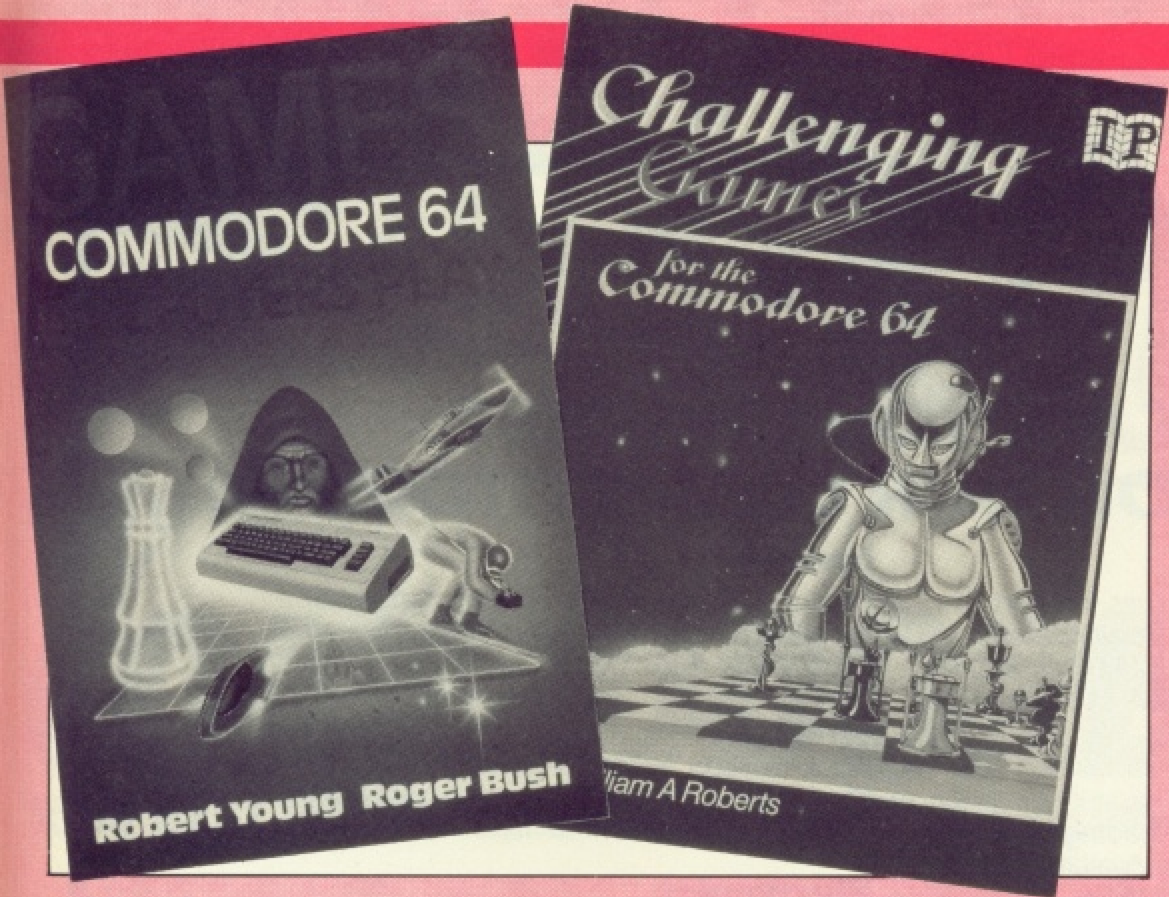
Book Title: Winning games on the Commodore 64
Authors: T.P. Barrett and S.W. Colwill
Publisher: Ellis Horwood Ltd.
Price: £5.95

From the title of this book, it isn't very clear whether the book provides "winning games" or whether it intends to help you win games. A quick scan of its contents shows that the book has a little more than the others of its genre. There are in fact two main sections, the second containing 21 reasonable albeit unexceptional games.

The most exciting bit is the first section which gives a good beginner's guide to the 64. The areas covered include BASIC, simple animation, outputting information to screen, Boolean operators, user defined graphics, sprites, joysticks routines and sound. The treatment of the subject matter was fair giving a good fund of information for the beginner. The section on sound was the exception in that it as was short and almost of no value at all. I got the impression that the authors preferred the graphical side of the 64. The chapter on the use of joysticks was of some value in as much that a machine code routine is provided for the scanning of the joysticks. This routine was in fact used in many of the games in section 2. On the whole the section is probably marginally better than the user's guide supplied with the 64.

The games listings provided were mainly "arcade" type games with the odd educational program. The instructions or guidance notes were minimal and the games were generally mediocre. I noted with some surprise that simulation or adventure games were included.

Overall, a rather disappointing book which falls badly between a games book and a user's guide and isn't really any good for either purpose.



Book Title: 41½ Fun projects for your Commodore
Author: D. Disharoon & H. Kohl
Publisher: Reston Publishing Company
Price: $6.95

This book is more of a variety of things to do rather than games to play and seems to be aimed at the younger end of the beginners market. The book is split into six sections covering word games, number games, IQ exercises, strategy puzzles, music programs and utilities. To my view, the strength of many of the routines given is that apart from being used as they stand, they can be used as a rich source of ideas and hints on programming techniques.

The most notable routines are a word search generator, a simple database, a tiny word processor, and several amusing puzzles. All programs had introductory texts describing the program and giving some indication on its operation.

If you want to simply play mindless zapping games, this book is not for you. If, on the other hand, you want to use the 64 and exercise your mind and programming skills, this book is a good starting point.

Book Title: Games Commodore 64 computers play
Authors: Robert Young & Roger Bush
Publisher: Addison-Wesley
Price: £6.95

At 179 pages, this book was one of the beefier of this collection. On opening the book it is pleasant to note that the book is split into discrete sections, each containing games of a particular type. (As the title indicates, this is a game only book).

Section one contains the ubiquitous "Arcade" games. As expected these compromise the usual space, chase and race types. A nice touch is that a number have both keyboard and joystick options. Overall a fair variety which should offer something of interest to most people.

Next come the simulations. Four such programs are given representing the main types. First there is a space adventure which, whilst being tolerably short, is reasonably challenging. Two of the games are of the "what if" genre where you change a number of parameters and hope that the result is correct. These games enable you to run a farm or control the Roman Empire. Finally, there is a war game simulating the Battle of Britain.

Last come the "mind games" in which you must try to think faster than a 6510. These games include 3D noughts & crosses, Reversi, and Nim. Not an inspiring collection but good honest staple stuff.

Each program has a piece of descriptive text which not only gives instructions on how to play the game but also indicates how sections of the programs function. The introduction has a full listing of the meaning of the CBM control characters and the graphics characters; invaluable if you aren't used to typing in listings.

Generally, a reasonably friendly text which should provide hours of harmless fun at an acceptable price.

Book Title: Challenging games for the Commodore 64
Author: W. A. Roberts
Publisher: Interface Publications
Price: £3.95

This volume presents a varied mix of 16 games. I got the overall impression that the games tended to require brains rather than reflexes. In fact the majority of the games were of the simulation, adventure or strategy type game. The most notable inclusions were Reversi and Chequers programs.

Rather an exciting book which is hardly better than average.

Book Title: The Commodore 64 Program Book
Author: Vince Apps Publishing Associates
Price: £4.95

This book contains twenty five programs of widely varying size and type including educational programs, games and utilities. Personally, I welcome this wide diversity since not all of us live for zapping aliens. Overall, the programs tend to be more suitable for the older user and this book is not suitable as a learning aid for young users.

All programs use BASIC and I note with interest that the author has acknowledged speed limitations of the language by offering a high proportion of speed independent programs. There are a number of adventure/simulation programs which, whilst being lengthy, will give long term entertainment. For the lovers of arcade type games, there are a fair proportion of "zapping" games. Inevitably, however, these games are all restricted by the use of BASIC.

For those seeking something more useful, there are several utilities including an assembler and a renumber routine. I consider the inclusion of the assembler to be a significant step forward. The program is quite versatile providing the ability to assemble, disassemble, save and load machine code and run the code. The book is almost worth buying for this program alone.

The quality of the book is good with all listings legible and with adequate instructions and program descriptions. At the price, well worth a long look.

Book Title: Crazy games for your Commodore 64
Authors: H. Renko & S. Edwards
Publisher: Addison-Wesley
Price: £3.95

Here we have a very handy little book of programs of all types. These authors have found an interesting blend of games and novelty programs. I was particularly pleased to see routines for the well known "Black Box" game and a program to generate Escher patterns. Overall there are thirty listings ranging from a rather amusing adventure to a Frogger look alike.

A tolerable amount of additional information is given with the programs, particularly where some theory of operation is required. Overall an excellent little book which represents good value and offers something for everyone.

YOUR COMMODORE

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
SALLY COLLINS
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

SOFTWARE APPLICATIONS

HOME ACCOUNTS. Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP inbuilt accuracy check. Records all transactions. Projects cashflow for any period ahead. Available for C.16 CBM64 or VIC-20. £8.45 or *free* details from Discus Software, Freepost, Windmill Hill, Brixham TQ5 9BR. Telephone 08045 55532.

LIBRARIES

COMMODORE 64
SOFTWARE LIBRARY
Over 200 titles. 2 weeks hire £1.00
Membership £5.00. Stamp for list.
Les Wilson (CY), 100 Blenheim Walk, Corby, Northants.

FOR HIRE

VIC-20/CBM 64
SOFTWARE HIRE
Free membership!!!
Dozens of titles (eg 30 Time Trek, Pharoah's Tomb) from 60p per week. Send S.A.E. for your Hire kit to:
Vic 20 Software Hire (Y.C.O.)
242 Ransome Road, Mapperley, Nottingham.
State which machine please.

SOFTWARE

TODDLERS!

Play with your parent's CBM 64. Animated musical flashcard alphabet and spelling program, developed in conjunction with school and educational psychologist. Suitable 1½-7 years.
Cassette **£5.00** Disc **£7.50**
Toddlersoft, 1 Catlane Cottage, Ewelme, Oxon OX9 6HX.
Full refund if not delighted.

COMMODORE 64 SOFTWARE

GAMES AND BUSINESS USE
Also new releases list every 2 weeks. Send your name and address to go onto our mailing list to:
M. J. Seaward (Dept Y.C.O.)
St Olaf's Road, Hill Head, Stratton, Nr. Bude, Cornwall.
Tel: Bude 4179 up to 10 pm.

To attract the buyers eye



PHONE 01-437 0699

ZOOMSOFT

COMMODORE 64 SOFTWARE SPECIALISTS

Title	cass	disc
Boulder Dash	8.95	10.95
Bristles	8.95	10.95
Astro Chase	8.95	10.95
Flip Flop	8.95	10.95
The Hulk	9.95	13.95
Fort Apocalypse	9.95	14.95
Aztec Challenge	8.95	12.95
Kong Strikes Back	7.90	—
Slinky	8.95	12.95
Zork I	—	11.95
Zork II	—	11.95
Zork III	—	11.95
Starcross	—	11.95
Deadline	—	11.95
Encounter	9.95	12.95
Flight Simulator	34.25	37.95
S.A.M.	—	41.95
Forbidden Forest	8.95	12.95
Jetboot Jack	7.94	—
Nato Commander	9.95	12.95
Sorceror of Claymorgue Castle	9.95	13.95
Trollie Wallie	6.95	8.95
Daley Thompson's Decathlon	7.95	—
Football Manager	7.95	—
Super Base 64	—	86.25
Juice	—	29.95
Adventure Creator	ROM	29.95
Oil's Well	ROM	26.95
Beyond Castle Wolfenstein	—	21.95

Title	cass	disk
Castle Wolfenstein	—	27.95
Master Type	—	32.75
Sprite Maker	9.95	—
Stranded	7.95	—
Letter Wizard	—	34.95
Mask of the Sun	—	28.75
Dril	—	25.75
Spell Wizard	—	34.95
B.C'S Quest for Tyres	9.95	—
Swag	8.95	10.95
Soldier Fortune	8.95	—
Zaga Mission	7.95	9.95
Ant Attack	8.95	—
Espionage	8.95	—
Mr Robot	8.95	11.95
Spitfire Ace	9.95	12.95
Beach Head	9.95	12.95
Dallas Quest	—	14.95
Bruce Lee	14.95	14.95
Knights of the Desert	14.95	14.95
Combat Leader	14.95	14.95
Tigers in the Snow	14.95	14.95
Caverns of Khafta	8.95	12.95
Blue Max	9.95	14.95
Multiplan	—	99.95
Paintbrush	ROM	19.95
Personal Accountant	25.60	—
Graphics Basic	—	25.55
Household Finance	22.85	—
Logo	—	56.25

100's more titles available. Send S.A.E. for free catalogue. Send cheques/POs to:

ZOOMSOFT
46 Huntsworth Mews, London NW1 6DB.
Tel: 01-723 0562

CLUBS

CLUB 64

For CBM64/1541 Users

If you are a CBM 64 user with disk drive you are invited to join CLUB64 the international user group.

On joining every member may order free of charge 3 disks from our excellent software library.

Annual membership - £12 UK and Ireland, £15 Mainland Europe and £20 all other countries.

Further details from:

BRENDAN CONROY, CLUB64, 85 Upper Drumcondra Road, Dublin 9, Ireland.

BUSICALC 3

- the sophisticated spreadsheet !

Easy to learn, easy to use - something that can't be said of many business programs. But it's true of all the programs in the BUSICALC series.

BUSICALC 3 can handle all sorts of jobs - budgets, expenditure analysis, stock lists, price lists, and product costing are just a few of the possibilities. Three-dimensional formulae automatically access data stored on disk, so that you can easily pull together information from several different sheets and summarise or manipulate it.

It's simple to transfer data to other programs such as Easy Script. And you can use virtually any printer with BUSICALC 3, whether dot matrix or daisy wheel, Commodore or non-Commodore.

For the CBM 64 and PET/CBM 4000 & 8000 series.

Available through dealers or from:
Supersoft, Winchester House, Canning Road, Harrow HA3 7SJ

Phone 01-861 1166 for more details and a free catalogue.

